Low Resource Efficient Speech Retrieval

by

Chunxi Liu

A dissertation submitted to The Johns Hopkins University
in conformity with the requirements for the degree of
Doctor of Philosophy

Baltimore, Maryland
October, 2018

Abstract

Speech retrieval refers to the task of retrieving the information, which is useful or relevant to a user query, from speech collection. This thesis aims to examine ways in which speech retrieval can be improved in terms of requiring low resources - without extensively annotated corpora on which automated processing systems are typically built - and achieving high computational efficiency.

This work is focused on two speech retrieval technologies, spoken keyword retrieval and spoken document classification. Firstly, keyword retrieval - also referred to as keyword search (KWS) or spoken term detection - is defined as the task of retrieving the occurrences of a keyword specified by the user in text form, from speech collections. We make advances in an open vocabulary KWS platform using context-dependent Point Process Model (PPM). We further accomplish a PPM-based lattice generation framework, which improves KWS performance and enables automatic speech recognition (ASR) decoding.

Secondly, the massive volumes of speech data motivate the effort to organize and search speech collections through spoken document classification. In classifying real-world unstructured speech into predefined classes, the wildly collected speech recordings can be extremely long, of varying length, and

contain multiple class label shifts at variable locations in the audio. For this reason each spoken document is often first split into sequential segments, and then each segment is independently classified. We present a general purpose method for classifying spoken segments, using a cascade of language independent acoustic modeling, foreign-language to English translation lexicons, and English-language classification. Next, instead of classifying each segment independently, we demonstrate that exploring the contextual dependencies across sequential segments can provide large classification performance improvements. Lastly, we remove the need of any orthographic lexicon and instead exploit alternative unsupervised approaches to decoding speech in terms of automatically discovered word-like or phoneme-like units. We show that the spoken segment representations based on such lexical or phonetic discovery can achieve competitive classification performance as compared to those based on a domain-mismatched ASR or a universal phone set ASR.

Primary Reader: Sanjeev Khudanpur

Secondary Reader: Hynek Hermansky

Acknowledgments

To Sanjeev Khudanpur, I am grateful to you for giving me the opportunity to do the PhD at Hopkins and thank you for your insights, patience, and constant support which made this work possible. I will always appreciate your guidance and mentorship. Under your wing, this has been the most productive period of my life thus far.

To Hynek Hermansky, I would like to thank you for graciously serving on several of my committees during the course of my PhD program, for your leadership in the Center for Language and Speech Processing, and for your advice and willingness to read through this thesis.

To Aren Jansen, thank you for introducing me to the speech research world. I am grateful to you for sharing your insights, enthusiasm and encouragement, and for your careful editing of our papers.

To Dan Povey, Jan "Yenda" Trmal, and everyone in the JHU Kaldi team, thank you for allowing me to be a part of your successes.

To Najim Dehak, Shinji Watanabe, Craig Harman, and everyone in the JHU LORELEI Speech team, thank you for the thoughtful advice, discussions, collaborations and remarkable accomplishments along the way.

To Ruth Scally and Debbie Race, thank you for always being ready to help,

your patience and abundant assistance through the years.

To Guoguo Chen, Puyang Xu, Keith Kintzley, Yuan Cao, Shuai Huang, Vijay Peddinti, Harrish Mallidi, Ming Sun, you have been very supportive and thank you for your guidance during the early years of my PhD. To Xiaohui Zhang, Matthew Wiesner, Pegah Ghahrmani, Vimal Manohar, David Snyder, Hang Lyu, and all my JHU friends and colleagues, thank you for being great company and being so willing to help me in times of need.

To my parents Huaijie and Guoyi, who have given me a very fortunate life, thank you for giving me so much love and support beyond what can be expected. Above all, this thesis is dedicated to you.

Dedication

This thesis is dedicated to my parents, Huaijie and Guoyi.

Table of Contents

List of Tables

List of Figures

List of Acronyms

AP	Average Precision
ASR	Automatic Speech Recognition
ATWV	Actual Term-Weighted Value
AUD	Acoustic Unit Discovery
BN	Bottleneck
CNN	Convolutional Neural Network
CV	Cross Validation
DARPA	Defense Advanced Research Projects Agency
DNN	Deep Neural Network
DTW	Dynamic Time Warping
EM	Expectation-Maximization
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Unit

G2P	Grapheme-to-Phoneme
HMM	Hidden Markov Model
IARPA	Intelligence Advanced Research Projects Activity
IL	Incident Language
IR	Information Retrieval
IV	In-Vocabulary
KWS	Keyword Search
LDA	Linear Discriminant Analysis
LDC	Linguistic Data Consortium
LORELEI	Low Resource Languages for Emergent Incidents
LSA	Latent Semantic Analysis
LSTM	Long Short-Term Memory
LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Maximum a Posteriori
MFCC	Mel-Frequency Cepstral Coefficient
MT	Machine Translation
NI	Native Informant
NIST	National Institute of Standards and Technology

NMI	Normalized Mutual Information
OOV	Out-of-Vocabulary
OTWV	Oracular Term-Weighted Value
PLP	Perceptual Linear Predictive
PPM	Point Process Model
RNN	Recurrent Neural Network
SDR	Spoken Document Retrieval
SF	Situation Frame
SGD	Stochastic Gradient Descent
STD	Spoken Term Detection
SVM	Support Vector Machine
TDT	Topic Detection and Tracking
tf-idf	term frequency-inverse document frequency
UTD	Unsupervised Term Discovery
VAE	Variational Autoencoder
WER	Word Error Rate
WFSA	Weighted Finite State Acceptor
WFST	Weighted Finite State Transducer

Chapter 1

Introduction

Nowadays the collections of audio data have been ever-increasing, including broadcast news, telephone conversations, meetings, lectures, etc, which can be referred to as "spoken documents". Storing and digitizing vast amounts of audio data is commonplace. In providing users with easy access to the information of their interest, information retrieval (IR) has been a growing area both in academia and in the market place.

> Given the user query, the key goal of the IR system is to retrieve information that is useful or relevant to the user. [1]

Early developments have focused on text IR, while the rapid growth of media sources such as audio, image and video has motivated the field of multimedia IR to support navigating large multi-media collections. Given the vast quantities of speech recordings, this thesis considers the practical pursuit of automatic information access to speech archives – speech retrieval.

> Speech retrieval refers to the task of retrieving the specific pieces of spoken audio data from a large collection that pertain to a query requested by a user. [2]

The basic application scenario assumes that a user translates their information need into a query and initiates the query, which can be a set of words in text or spoken form, and the system will return either a list of rank-ordered documents, or any specific document segments that are relevant to the query. We also consider speech retrieval and spoken content retrieval [3] as synonymous, which includes the tasks of spoken document retrieval, spoken term detection or keyword search, topic detection[1], etc., and we will discuss these tasks in more detail in subsequent Section 1.1.

Speech retrieval can be approached in ways lying between information retrieval and automatic speech recognition (ASR), while it can be challenging to build high-accuracy ASR systems in real-world scenarios, due to the diversity of languages and the requirement of extensively annotated corpora on which the ASR algorithms are typically built. Additionally, the fact that the audio data volumes are ever increasing has posted requests for any algorithm design on the progress of time and space efficiency. Accordingly, these challenges have led to various speech retrieval techniques beyond cascading ASR with text IR [3]. The goal of this thesis is to further improve speech retrieval techniques given the language diversity and low human annotation resources.

[1]The notion of topic here can be considered as a general cluster or class. The topic detection task may refer to either an unsupervised learning problem like document clustering, or a supervised learning task such as document classification. The supervised document classification is also usually referred to as document categorization, topic classification, topic identification, etc. [4].

1.1 Background and motivation

This section provides an overview of a few large-scale evaluations or programs that have led to significant progress on speech retrieval technologies, and describes the posed challenges that motivate this thesis.

The National Institute of Standards and Technology (NIST) sponsors an annual Text REtrieval Conference (TREC), which was started in 1992, to encourage research on information retrieval and provide infrastructure for large-scale text retrieval evaluations. A series of past evaluations, TREC-6 – TREC-8, began to include a Spoken Document Retrieval (SDR) track, where systems are posed with queries and attempt to return a list of documents ranked by decreasing similarity to the queries [5]. Given the sufficiently accurate ASR on broadcast news speech, SDR was considered to be a "solved" problem [5].

NIST has also run a series of evaluations on another speech retrieval task, Topic Detection and Tracking (TDT), since 1998 [6]. TDT includes five tasks named, Topic Tracking, Link Detection, Topic Detection, First Story Detection and Story Segmentation. TDT was made multilingual by expanding the corpora to include broadcast news of English, Mandarin, and Modern Standard Arabic. TDT research in general aims to develop algorithms for detecting new topics in streams of broadcast news, and then tracking these topics over time. Notable generalizations arised from evaluation, e.g., cross-lingual processing performance degraded compared to monolingual processing [7]. TDT along with multilingual modeling have been popular research problems since then.

Thus far both SDR and TDT focused on the broadcast news domain. Instead NIST ran another speech retrieval evaluation – 2006 Spoken Term Detection (STD) Evaluation – specifically towards automatically detecting the occurrences of each given term from audio corpora of heterogeneous speech material [8]. Compared to processing broadcast news, searching spontaneous conversational speech posed more challenges and raised pragmatic awareness of domain robustness, system scalability, out-of-vocabulary (OOV) queries [9], etc. In addition, we see significant performance degradation on Arabic and Mandarin data as compared to English [8], and such markedly lower performance on non-English languages requires developing more effective language independent solutions.

Following the 2006 STD evaluation, the Intelligence Advanced Research Projects Activity (IARPA) conducted the Babel Program [10] starting in 2011, of which the goal is to develop scalable multilingual keyword search (KWS) capabilities that can be rapidly applied to any human language. However, most of the world's languages lack the large amount of manually-transcribed, manually-translated, or manually-annotated corpora that the standard automated algorithms strongly rely on, and these languages can be referred to as underserved or low resource languages. Particularly in developing ASR and KWS systems for underserved languages, collections of correponding transcribed speech or phonetic lexicons can be severely limited. Whereas significant progress has been made in automatically recognizing and searching underserved languages, by exploiting various novel techniques such as semi-supervised training of neural network-based acoustic models [11], building

language independent acoustic model through sharing a common phone set [12], learning multilingual neural network-based acoustic features [13], etc.

Also concerned with advancing human language technology performance for underserved languages, the ongoing Low Resource Languages for Emergent Incidents (LORELEI) Program supported by the Defense Advanced Research Projects Agency (DARPA) introduces a Situation Frame (SF) task, which aims to retrieve and aggregate information from text and speech documents in the context of emergent situations – such as natural disasters or disease outbreaks – in locations where low-resource languages are spoken [14, 15]. The relevant documents and associated situational information are collectively referred to as situation frames (SFs), and each SF consists of the situation type (also simply referred to as topic), geographic localization, and situation status. Retrieving SFs from speech can be formulated as component tasks including topic identification and keyword search [16]. The retrieved SFs are intended to provide situational awareness for emergent missions such as humanitarian assistance or disaster relief operations.

In general, we can see a great deal of interest in expanding speech retrieval coverage of the world's languages, while in many cases the resources available to build the typical automated processing systems, i.e., manually-transcribed speech or any manual linguistic annotations (e.g., topic labels per document), are severely limited. Therefore, extensive research efforts on improving various speech retrieval techniques have thus far been focused on developing language-independent and scalable solutions that require zero or low manually-annotated resources for the language of interest, which are also

the motivating problems of this thesis.

1.2 Problem statement

This thesis touches on three individually challenging tasks that can serve as basis technologies for speech retrieval – keyword retrieval, automatic speech recognition (ASR), and spoken document classification.

1.2.1 Keyword retrieval

In this work we consider spoken keyword retrieval, spoken term detection (STD) and keyword search (KWS) as synonyms. Typically, the input speech audio waveform is converted into a sequence of fixed-dimensional acoustic vectors $\mathbf{O} = \mathbf{o}_1, ..., \mathbf{o}_T$, by a process called feature extraction. Consider we have a fixed vocabulary set $\mathcal{V}$, and we treat a sentence as composed of a sequence of words $\mathbf{W} = w_1, ..., w_N$, where each w_i, for $i = 1 \dots N$, corresponds to a word type $w \in \mathcal{V}$. Then the keyword retrieval task can be stated as follows. Given acoustic observations $\mathbf{O}$, assume a word type w is a keyword of interest (in written form in the native orthography), and each occurrence of word w_i is defined as a triplet (w, t_b, t_e), where w is the word type, t_b is the beginning time of this word occurrence and t_e is the end time. Spoken keyword retrieval, STD, or KWS is to find all the occurrences of a keyword type w in acoustics $\mathbf{O}$. Alternatively, the query keyword can also be a contiguous sequence of words, i.e., word n-grams with $n \geq 2$, and keyword retrieval is to find occurrences of the same n-grams.

Development of such technique can provide speech retrieval system with

the functionality that, when the user enters a query keyword, the system can return the utterances containing the keyword, or the exact occurrence time spans of the keyword.

1.2.2 ASR

Given each speech utterance parameterized as $\mathbf{O} = \mathbf{o}_1, ..., \mathbf{o}_T$, ASR is to find the underlying word sequence $\mathbf{W} = w_1, \ldots, w_N$. The statistical formulation can be expressed as:

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}} P(\mathbf{W}|\mathbf{O}) \tag{1.1}$$

where optimal word sequence $\hat{\mathbf{W}}$ is the one most likely to have generated the acoustic sequence $\mathbf{O}$. We can apply Bayes' rule to decompose the posterior probability $P(\mathbf{W}|\mathbf{O})$ as:

$$P(\mathbf{W}|\mathbf{O}) = \frac{P(\mathbf{O}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{O})} \tag{1.2}$$

where $P(\mathbf{O}|\mathbf{W})$, the observation likelihood, is computed by the acoustic model, and $P(\mathbf{W})$ is the prior probability, computed by the language model. Each major ASR component – i.e. acoustic model, pronunciation lexicon (as a mapping from words to phoneme strings), and language model – can be formulated as probabilistic model, and often be represented by weighted finite state transducer (WFST) [17, 18]. Common methods for combining and optimizing probabilistic models in ASR can be efficiently implemented by the well-defined operations on WFSTs [17]. Thus, the individual ASR components can be integrated and processed into a single WFST, which represents the composed probabilistic model and is referred to as decoding graph [18, 19].

Given the acoustic observations, searching through all the possible word sequences and finding the one which has the highest posterior probability is referred to as the *decoding* problem in ASR. In the WFST framework, we construct an acceptor (or WFSA) for each speech utterance, compose the acceptor with decoding graph, and obtain a *search graph*, called S [18]. Then decoding is equivalent to finding the best path through S, e.g. by the Viterbi algorithm [20]. In practice, we generate a pruned subset of S, by a process of lattice generation, and find the best path through the subset. A lattice is an acyclic directed graph that efficiently represents multiple ASR hypotheses, i.e. a WFSA with word or phoneme labels.

ASR can be used to convert speech data into plain text, to which standard text-based retrieval can be applied. However, given the suboptimal ASR accuracies in many realistic cases, the one-best ASR transcription may have low recall rates for the important query-relevant words. Instead the efficacy of indexing ASR lattice has been demonstrated to improve various speech retrieval tasks, e.g., spoken document retrieval [21] and spoken term detection [9]. Thus how to efficiently generate ASR lattice, with an optimal trade-off between a compact lattice size and decoding speed, is also an important line of research [18, 19].

1.2.3 Spoken document classification

Consider a collection of spoken documents represented as $\mathcal{X} = \{x_1, \ldots, x_{|\mathcal{X}|}\}$, where each data instance $x_i \in \mathcal{X}$, for each $i = 1 \ldots |\mathcal{X}|$, denotes one document. The closed set of predefined classes is denoted as $\mathcal{K} = \{1, ..., k, ..., |\mathcal{K}|\}$. Each

document or data instance x_i is associated with a $|\mathcal{K}|$-dimensional binary class vector $\mathbf{y}_i$ where each dimension k, for each $k = 1 \ldots |\mathcal{K}|$, is assigned to 0 or 1, and specifies if that class k applies to x_i. The document classification is to find the corresponding binary class vector $\mathbf{y}_i$ for each x_i.

The 'class' mentioned above is also referred to as category, label, or topic. 'Topic' typically corresponds to the notion of a discourse subject, while in this work we consider a topic as equivalent to a general document class. Thus we simply consider document classification, document categorization, topic classification, and topic identification as equivalent. In addition, document classification is defined as single-label classification if there is always only one class that applies to each x_i (i.e. there can only be a single 1 in each class vector y_i), and as multi-label classification if there can be an arbitrary number of classes for each x_i.

Document classification has been a key technology in information retrieval nowadays [1]. Classification using standing queries can organize document collections and retrieve the relevant ones, e.g., routing or filtering emails/voicemails for their own purposes [4], identifying incident-related audios and emerging needs therein for disaster response planning [15], etc.

1.3 Contributions

The overall contributions of this thesis take a step towards language independent and scalable speech retrieval capabilities. In particular we focus on improving two specific tasks – spoken keyword retrieval and document classification – in support of the overall goal, and the respective set of contributions

to each task are enumerated below.

Spoken keyword retrieval with point process modeling. The original presentation of the point process model (PPM) for keyword search in [22] detailed the theoretical development, and the subsequent series of works have improved the model estimation and search algorithms [23, 24]. The first set of contributions of this thesis begins with the demonstration that PPM framework provides the state-of-the-art OOV keyword search performance, and posts substantial fusion gains when combined with hidden Markov model (HMM) based keyword search outputs. In light of the phonetic variations across differing contexts, the next contribution extends the PPM framework to operate on context-dependent phonetic event patterns instead of monophone streams considered in the past. The final contribution in this line of work is the accomplishment of a PPM-based lattice generation framework that enables both keyword search and ASR decoding. We demonstrate that combining context-dependent point process modeling and detection-based lattice generation yields significant improvements in keyword search performance compared to the prior monophone-based PPM approach.

Spoken document classification with language-independent ASR. Audio documents collected in the wild may be extremely long and contain variable class label shifts at variable locations in the audio, so each audio document needs to be split into a sequence of speech segments, and then each resulting segment can be individually classified into predefined classes. The next set

of contributions in this thesis first explores a general purpose approach for classifying speech segments, using a cascade of language-independent acoustic modeling, foreign-language to English translation lexicons, and English-language classifiers. Next, instead of classifying each segment independently, we develop contextual classifiers that additionally encode context dependencies across adjacent segments. While both recurrent neural network and attention network based approaches can provide performance improvements, the proposed position-aware attention network that allows for using contexts via a selective manner can consistently outperform the context-independent classifiers.

Spoken document classification with unsupervised speech technologies. The final set of contributions first address the requirement of an acoustic model in absence of any orthographic lexicon. We exploit unsupervised lexical and phonetic discovery approaches to inferring the lexical and phonetic inventory of a language, via dynamic time warping based unsupervised term discovery and Bayesian acoustic unit discovery (AUD), respectively. We extend a prior deep generative AUD framework – structured variational autoencoder (VAE) – to a structured *context-sensitive* VAE with a hybrid feedforward encoder and a recurrent decoder, which achieves state-of-the-art AUD performance. Next, we demonstrate that the bag-of-words representations based on the automatically learned units from either lexical or phonetic discovery can provide competitive classification performance when compared with those based on the word hypotheses from language-independent ASR. Lastly, given the

acoustic unit sequences, we develop a convolutional neural network based representation and classification framework, and show, when given sufficient classification training data, it can significantly outperform the bag-of-words representation.

1.4 Outline

The rest of this thesis is organized as follows. Chapter 2 examines how to improve spoken keyword search based on point process modeling. We begin by briefly describing the prior work of the point process model for keyword search. Then we discuss how to apply PPM to low-resource settings where the amount of transcribed speech is severely limited and the pronunciation dictionary is incomplete. We subsequently present how to perform context-dependent point process modeling and how to generate the ASR lattice in the PPM framework. In all cases we evaluate PPM performance in the IARPA Babel Program framework.

Chapter 3 and 4 are focused on classifying spoken documents, or classifying spoken segments if each document needs to be first split into segments, where very small amount (minutes rather than hours) or even none of transcribed speech is available to train an ASR system in the language of interest. Chapter 3 explores a general method of using a language-independent acoustic model through sharing a common phonemic representation across languages, translating ASR transcripts to English, and then applying an English classifier. We first benchmark the performance of context-independent classifiers on the LORELEI datasets, where each spoken segment is classified independently.

Then we develop context-dependent classifiers, where to classify each segment its context segments need to be considered.

In Chapter 4, we shift our focus from supervised training of ASR systems – i.e. using any available transcribed speech and orthographic lexicons – to unsupervised learning of acoustic models. We begin with introducing unsupervised term discovery (UTD) and acoustic unit discovery (AUD). Then we present how to develop a deep generative AUD framework with structured context-sensitive variational autoencoder, and evaluate the automatically discovered acoustic unit sequences against the orthographic phoneme transcripts on TIMIT and Switchboard corpus. Next, given the acoustic unit sequences, we present a convolutional neural network based framework in comparing the bag-of-words document representation. Finally, we perform comprehensive topic classification evaluations on the LORELEI datasets using outputs from UTD, AUD and ASR.

In Chapter 5, we summarize the developed individual components and their connection to the improved speech retrieval technologies, and discuss possible directions for future work.

1.5 Related publications

Large portions of Chapter 2, 3, and 4 have appeared in the following papers:

1. Chunxi Liu, Aren Jansen, Guoguo Chen, Keith Kintzley, Jan Trmal, and Sanjeev Khudanpur, "Low resource open vocabulary keyword search using point process models," in *Proceedings of Interspeech*, 2014.

2. Chunxi Liu, Aren Jansen, and Sanjeev Khudanpur, "Context-dependent point process models for keyword search and detection-based ASR," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

3. Chunxi Liu*, Jinyi Yang*, Ming Sun, Santosh Kesiraju, Alena Rott, Lucas Ondel, Pegah Ghahremani, Najim Dehak, Lukas Burget, and Sanjeev Khudanpur, "An empirical evaluation of zero resource acoustic unit discovery," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. (Both authors contributed equally.)

4. Chunxi Liu, Jan Trmal, Matthew Wiesner, Craig Harman, and Sanjeev Khudanpur, "Topic identification for speech without ASR," in *Proceedings of Interspeech*, 2017.

5. Matthew Wiesner, Chunxi Liu, Lucas Ondel, Craig Harman, Vimal Manohar, Jan Trmal, Zhongqiang Huang, Najim Dehak, and Sanjeev Khudanpur, "Automatic speech recognition and topic identification for almost-zero-resource languages," in *Proceedings of Interspeech*, 2018.

6. Chunxi Liu, Matthew Wiesner, Shinji Watanabe, Craig Harman, Jan Trmal, Najim Dehak, and Sanjeev Khudanpur, "Low-resource contextual topic identification on speech," in *Proceedings of the IEEE Spoken Language Technology (SLT) Workshop*, 2018.

Chapter 2

Spoken Keyword Retrieval with Point Process Models

The goal of this chapter is to develop scalable multilingual keyword search (KWS) capabilities with limited access to the typical linguistic resources that state-of-the-art speech recognition technologies strongly rely on. The dominant mode of the KWS research thus far has been adapting the high-resource large-vocabulary continuous speech recognition (LVCSR) based keyword search systems that were developed for the NIST 2006 Spoken Term Detection evaluation [8] to this low-resource setting. However, with the present restricted availability of transcribed speech for language model estimation and highly incomplete pronunciation lexicons producing high keyword OOV rates, the main strengths of LVCSR for search are substantially handicapped. These programmatic constraints thus provide an opening for previous-generation lightweight phonetic search methods to play a continued role.

2.1 Introduction

Originally presented in [22], the point process model (PPM) for keyword search is a whole-word acoustic modeling and search technique. The PPM is founded on the hypothesis that the timing of robustly identifiable phonetic events provides sufficient cues to decode the underlying linguistic message, which in the present case are occurrences of a given keyword. The PPM trades pronunciation-derived hidden Markov modeling of frame-level phonetic likelihoods for inhomogeneous Poisson process rate parameters characterizing the likelihoods of phonetic event arrivals throughout the keyword. A series of past efforts have been focused to improve the model estimation and search algorithms [25, 26, 23]. Past studies have demonstrated that sparse phonetic event-driven PPMs permit unprecedented speeds in search collection indexing [24] and improved robustness to noise [27]. Moreover, in high-resource settings the PPM was demonstrated to outperform competing phonetic fast lattice search methods in both search speed and accuracy [24].

In Section 2.3 of this chapter[1], we consider the application of PPM-based keyword search technology to the low-resource multilingual setting. To participate in this challenge space, we consider multiple extensions to the basic framework. First, like HMM-based lexical models, the PPMs require a frame-level phonetic acoustic model to generate the phonetic event streams. Thus, we evaluate PPM performance in conjunction with a truly state-of-the-art deep neural network (DNN) acoustic model tailored to the present low resource setting. Second, the original PPM framework required keyword

[1] Large portions of this chapter have been published in [28, 29].

training examples to estimate Poisson rate parameters, while the recently proposed maximum a posteriori (MAP) estimation technique allows back-off to a dictionary-derived prior [23]. Given the present preponderance of out-of-vocabulary keywords (which are also out-of-training), we evaluate the use of a grapheme-to-phoneme conversion tool to seed dictionary-based PPMs. Additionally, to evaluate LVCSR search complementarity, for the first time we consider the system combination potential of our PPM keyword search system.

However, the past comprehensive benchmark evaluations have thus far been limited to building the PPM search index and parametric models on monophone event patterns without considering the phonetic variations across differing contexts, in contrast to common practices employed by context-dependent (triphone) HMM-based ASR systems [30]. [27] is the only related work of using acoustic event patterns beyond monophone detectors, where untied states of whole-word GMM-HMM acoustic models were used to define the detector set. However, that work considered only a small vocabulary digit recognition task that required many examples of each word in the lexicon. In Section 2.4, we exploit DNN acoustic models to generate the tied triphone state (senone[2]) events, which enable the application of dictionary-based PPMs and subsequent MAP estimation for scaling to open vocabulary search tasks.

In addition to open vocabulary search, we also consider in Section 2.5 the use of our context-dependent PPMs for LVCSR, which is possible due to

[2]Senone refers to the tied triphone HMM state after the tree-based HMM state clustering [30], and it is also used as the neural network output unit in the hybrid DNN-HMM acoustic model [31].

recent advances in the computational efficiency of PPM search algorithms. We employ the detection-based ASR framework previously considered for small vocabulary tasks [32, 27]. In contrast to the Viterbi search of HMM systems, this alternative approach applies a set of parallel word detectors and derives the most likely word sequence from their combined output. Critical to this process is the construction of a word lattice from the set of independent word detections so that language models can be subsequently applied. We first adapt the confusion network [33] algorithm as our baseline approach and propose our own lattice construction algorithm specially designed for the PPM framework. Both data structures can be then composed with a finite state transducer (FST) based language model and either decoded for LVCSR or used as the keyword search index for in-vocabulary queries.

Finally, in Section 2.6 we evaluate our proposed approaches with comprehensive KWS and LVCSR experiments under the IARPA Babel Program framework [10, 34], which aims to develop robust low-resource techniques to facilitate KWS search on massive multilingual speech corpus. We find the PPM system reaches state-of-the-art OOV search performance at a small computational cost. Moreover, we show that due to their complementary methodologies, combining PPM outputs with the LVCSR baseline produces substantial performance improvements. Finally we find incorporating context-dependency into the PPM framework produces large improvements over the original monophone PPM system and demonstrates reasonable LVCSR performance with a small computational footprint.

2.2 The Point Process Model for Keyword Search

In this section we begin with a brief review of the point process model for keyword search.

2.2.1 Poisson process models

Originally proposed in [22], the PPM for KWS is a parametric approach that assumes observed phonetic events derived from the input speech signal are generated by underlying keyword-specific Poisson processes. The PPM KWS framework first transforms input speech signals into smoothed phone posteriorgram[3] trajectories. Each phonetic event, which corresponds to a single phone occurrence, is subsequently selected as the local maxima of the smoothed posterior trajectories exceeding a threshold [25], which distills dense frame-level phonetic likelihood estimates into a minimal set of discrete phonetic sequences in time. This collection of extracted phonetic events provides the phonetic index of the search collection. Formally, given a time interval $(t, t+T]$, for each phone p in phone set $\mathcal{P}$, we denote its phonetic event set in time at which phone p occurs relative to time t as $N_p = \{t_1, \ldots, t_{n_p}\} = \{t_i\}_{i=1}^{n_p}$, where n_p is the total number of events within $(t, t+T]$ for phone p. Then the set of all observed events arriving in $(t, t+T]$ is $O_{t,t+T} = \{N_p\}_{p\in\mathcal{P}}$.

Thus given a keyword w with its occurrence time t and duration T, $O_{t,t+T}$ denotes the set of observed phonetic events during the course of a given word utterance. The arrival of phonetic events during the word realization is

[3]Phone posteriorgram refers to each phone posterior probability across a phone set as a function of time.

modeled as a collection of inhomogeneous Poisson processes, one per phone. We approximate the continuous Poisson rate function in interval $(t, t+T]$ as a piecewise constant function over D uniformly spaced divisions in $(t, t+T]$, with the inhomogeneous rate parameter for phone p denoted as $\lambda_{p,d}$ for each $d = 1, ..., D$. Also we make a corresponding subdivision in each phonetic event set N_p into D equal-size partitions [35] such that, $\forall d = 1 \dots D$,

$$N_{p,d} = \{t_i \in N_p | t_i \in (t + (d-1)\Delta T, t + d\Delta T]\} \tag{2.1}$$

where $\Delta T = T/D$, and accordingly

$$n_{p,d} = |N_{p,d}| \tag{2.2}$$

We denote the set of keyword-specific model parameters as θ_w, and thus the likelihood of the entire collection $O_{t,t+T}$ under θ_w given T can be expressed as

$$p(O_{t,t+T}|T, \theta_w) = \prod_{p \in P} \prod_{d=1}^{D} (\lambda_{p,d})^{n_{p,d}} e^{-\lambda_{p,d}\Delta T} \tag{2.3}$$

The PPM framework makes the assumption that the phonetic event timing distributions are independent of the candidate word duration T, and linearly scales all arrival times in $(t, t+T]$ onto the interval $(0, 1]$ to generate the transformed event set $O'_{t,t+T}$. Thus, after a change of variables, the likelihood function of Eq. 2.3 with $O'_{t,t+T}$ becomes

$$p(O'_{t,t+T}|T, \theta_w) = \prod_{p \in P} \prod_{d=1}^{D} (\lambda_{p,d})^{n_{p,d}} e^{-\lambda_{p,d}/D} \tag{2.4}$$

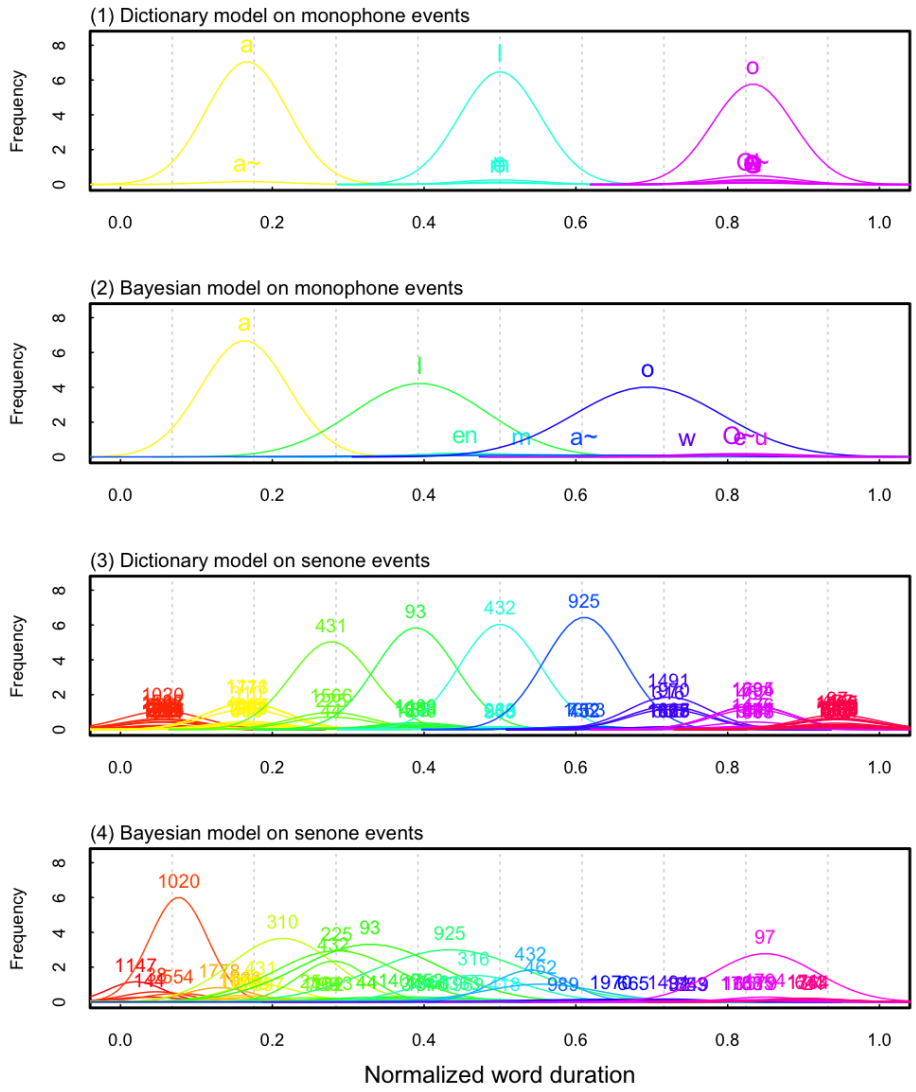


Figure 2.1: Dictionary/Bayesian MAP estimated phone timing models for the keyword “alo", based on monphone/senone events.

The phonetic event distribution (i.e., the time-varying Poisson rate function) of each phone instance within a word can be modeled by a single Gaussian distribution, and given the dictionary, a PPM can be constructed by assigning a Gaussian to each phone in the pronunciation [23]. Each Gaussian is further transformed to a mixture of Gaussians (GMM) to account for phone

confusions, where the mixture weights can be estimated over entire corpus (by aligning the observed/estimated phonetic events and the true phoneme transcriptions). For example, a dictionary model for the Haitian word “alo" is shown in Figure 2.1(1). Given the phonetic pronunciation of each keyword, a PPM can be constructed entirely based on the phonetic pronunciation provided by a dictionary.

Further, the GMMs – i.e., mixture weights, means and variances – can be updated by maximum a posteriori (MAP) estimation, benefiting from the observed phonetic event timing information of any available training examples [23]. This MAP estimate enables the dictionary model to fold in the observed event timing patterns of any available word exemplars present in the training corpus. As an illustration, the resulting MAP updated model for “alo" is depicted in Figure 2.1(2).

The PPM also requires a background model for likelihood normalization; here, we assume that outside the keyword of interest, phonetic events are generated by a homogeneous Poisson process characterized by a single independent rate parameter μ_p for each phone p. Thus, the likelihood of observation $O_{t,t+T}$ under the background model with parameters θ_{bg} is obtained as

$$p(O_{t,t+T}|T,\theta_{bg}) = \prod_{p\in P}(\mu_p)^{n_p}e^{-\mu_p T} \tag{2.5}$$

2.2.2 Point process model detection function

To evaluate an unknown utterance, we define the keyword detection function $d_w(t)$ as the log-likelihood ratio of phonetic events as described under the keyword and background model. This takes the form

$$\begin{aligned} d_w(t) &= \log \left[\frac{P(O_{t,\infty}|\theta_w)}{P(O_{t,\infty}|\theta_{bg})} \right] \\ &= \log \left[\int_0^\infty \frac{p(O'_{t,t+T}|T,\theta_w)P(T|\theta_w)}{T^{|O_{t,t+T}|}p(O_{t,t+T}|T,\theta_{bg})} dT \right] \\ &\approx \max_T \log \left[\frac{p(O'_{t,t+T}|T,\theta_w)P(T|\theta_w)}{T^{|O_{t,t+T}|}p(O_{t,t+T}|T,\theta_{bg})} \right] \end{aligned} \tag{2.6}$$

where the hypothesis keyword duration T serves as a latent variable with $P(T|\theta_w)$ modeled by a gamma distribution. For each keyword w we estimate a discrete set $\mathcal{T}$ that has a number of candidate durations. The integral can be approximated by computing over each candidate duration $T \in \mathcal{T}$, and taking the max (with the corresponding T as the hypothesized duration) [26]. The detection function is evaluated at each t, and a keyword detection is declared at each local maximum of $d_w(t)$ above a given threshold.

2.3 Low-resource open vocabulary KWS with PPM

In this section we describe the individual components of our low-resource PPM recipe.

2.3.1 Deriving phonetic events from low-resource DNNs

Over the past few years, DNN-HMM hybrid acoustic modeling has been widely used in state-of-the-art speech recognizers. One of our present goals is to evaluate these acoustic models in the PPM framework based on the assumption that the published word error rate reductions will translate into more accurate phone posterior estimates and, in turn, more accurate phonetic event streams. Now, one of the primary innovations relative to earlier waves of neural networks for ASR is the use of context-dependent HMM state targets. To use these DNNs in the PPM framework, we need to derive monophone posteriorgrams to enable the extraction of the requisite phonetic events. This is easily accomplished by summing together the posterior trajectories of HMM states corresponding to the same context-independent center phone. While we use the DNN trained in the context of an LVCSR system, once we derive monophone posteriorgrams our processing diverges completely from the HMM models and finite state machine based decoders.

Compared with the past neural network phonetic acoustic models [22, 24] evaluated in the PPM framework, our implementation introduces three new components. First, our DNN is trained on top of acoustic features that are speaker adapted with constrained maximum likelihood linear regression (CMLLR), also known as feature-space MLLR (fMLLR) [36]. Note that during training, fMLLR transform estimation is done through computing training alignments using a standard GMM-based, speaker adaptively trained model; in decoding, fMLLR transforms are obtained through first-pass decoding.

Thus, for both training alignments and first-pass decoding, the entire knowledge of phonetic context-dependency, pronunciation lexicon and word-level grammar will be integrated, which is absent from the previously employed phoneme recognition system that use monophone classes as prediction targets [37]. Second, in addition to basic Perceptual linear predictive (PLP) [38] features, we add pitch and probability of voicing (POV) features via the pitch extraction algorithm described in [39]. Experiments in [39] demonstrate that these pitch and POV features give substantial performance improvements on both tonal and non-tonal languages for LVCSR system, which also contributes to better estimation of phone posteriors. Finally, given the recent success of generalized maxout nonlinear activation functions in DNN modeling, we rely on a DNN acoustic model with p-norm activations [40] of the form $y = \|\mathbf{x}\|_p = (\sum_i |x_i|^p)^{\frac{1}{p}}$, where $\mathbf{x}$ represents a group of neuron inputs. Experiments in [40] demonstrate that DNNs using p-norm units with $p = 2$ perform consistently better than various other nonlinearities evaluated in speech recognition tasks, especially in low-resource conditions.

2.3.2 Searching for out-of-vocabulary keywords

We consider the KWS task in which keywords are provided in written form in the native orthography and a pronunciation lexicon is given with fixed vocabulary. However, in the low-resource setting a typical condition is that the pronunciation of a given keyword is not covered in the available lexicon. In this case, for the phonetic-based KWS system one standard solution is to predict the pronunciation of OOV keywords by using grapheme-to-phoneme

(G2P) conversion [41]. Thus, all OOV keywords become in-vocabulary (IV) and the updated lexicon would contain the phonetic composition of all keywords. However, in many applications or evaluation frameworks, redecoding the search collection is not applicable or allowed after the keywords are known, so other means are required to search using these new predicted pronunciations. Recently, a novel OOV processing technique called proxy keyword search [42] was demonstrated to produce state-of-the-art performance for the task. This method uses the G2P pronunciations of OOV keywords to generate a list of likely-confusable proxy words from the vocabulary. Using a cascade of weighted finite state transducer compositions with the original LVCSR lattice produces putative hits of the OOVs along with lattice posterior confidence scores. Proxy keyword search serves as the baseline OOV method in our experiments.

Using the MAP estimation framework of [23] and given a phonetic pronunciation for an OOV keyword produced by the G2P system, we can construct the dictionary prior PPM. Since we have no examples to estimate the Gaussian parameters within an OOV keyword, we can either assign Gaussian means at equal intervals with fixed variance (based on the simplifying assumption that all phones within the word have equal duration) [23], or estimate the Gaussian parameters for each phone using average phone durations [43]. In this chapter, we limit our evaluation to the simple uniform approach, though we would expect the incorporation of average phone duration statistics to provide marginal gains. We further introduce additional Gaussians of likely confused phones that are not in dictionary form using a confusion matrix

estimated across entire corpus. Moreover, we apply the Monte Carlo sampling approach explained in [24] to estimate Gamma distribution parameters of each keyword duration model for unseen words. In this way, we can construct a reasonably accurate estimate of PPM rate and word duration parameters without any training exemplars.

2.3.3 System combination

We evaluate the combination of the LVCSR and PPM search results by merging the respective putative hit lists. Both system use the identical DNN acoustic model but generate search ranked lists using completely different lexical models and decoding methodologies. The LVCSR system applies HMM lexical models on top of DNN-derived emission likelihoods in a WFST-based decoder that uses a language model. It generates deep word-based lattices that form the search index used for both IV and OOV keywords. The PPM system processes posteriors into an extremely sparse phonetic index and performs a linear-time search. Thus, the system combination evaluation serves to measure the complementarity of these techniques *after* the acoustic processing stages. The resulting putative hit lists from two systems are combined by the following procedure. First, we perform separate score normalization for each using the term-specific threshold technique in [44]. Second, we merge the hits from the two lists that begin and end with less than 0.5 second difference. The combined score for merged hits s_{merge} is computed as

$$s_{\text{merge}} = (w_1 s_1^{1/r} + w_2 s_2^{1/r})^r \tag{2.7}$$

where s_1 and s_2 are the individual system scores, w_1 and w_2 are the weights assigned to each system such that $w_1 + w_2 = 1$, and r is a power factor between 1 and 10. The parameters $\{w_i\}$ and r are optimized on a development set. Note that given 0-1 normalized input scores, this nonlinear combination rule will produce 0-1 normalized combination scores. Finally, we apply score normalization to the merged hit list.

2.4 KWS with context-dependent PPM

This section describes how we extend the PPM framework to operate on context-dependent phonetic event patterns instead of the previously used monophone patterns.

2.4.1 Deriving context-dependent phonetic events from DNN

To generate the context-dependent phonetic event streams, we use the DNN acoustic model as described in Section 2.3.1. We take as our events the set of tied triphone HMM states (senones), which are derived from traditional decision tree clustering of triphone states [30]. The DNN forward pass produces posteriorgrams over the senones which provide the input to the PPM pipeline described above, but where the monophone category set $\mathcal{P}$ is now replaced with the set of senones. The PPM search index is created by filtering the posteriorgrams according to the empirical distribution of each senone's duration and extracting the local maxima exceeding an empirically assigned threshold [25].

2.4.2 Context-dependent PPM construction

The original dictionary PPM is constructed by the monophone sequence provided by the pronunciation lexicon, so now we need to extend the dictionary form to that based on triphones, and construct the dictionary PPM based on the senone sequence. Given the left and right context phones, we can obtain the senone index for each central phone by answering the questions in phonetic decision tree. However, for the first and last phones of a single keyword the left and right context phones, respectively, are unknown without identifying the adjacent words. Thus, we assume that each phone in the phone set is equally likely to be the unknown context phones and we accumulate the senone index count by considering all these possibilities. We normalize each senone index count to determine the senone probability that is subsequently used as the GMM mixture weight in that position. Finally, we smear and renormalize the mixture weights using a global senone confusion matrix estimated from the training corpus.

The resulting dictionary PPM of word "alo" consisting of senones indexed by integers is shown in Figure 2.1(3). MAP estimation including any training instances of the word is subsequently performed using the observed senone event streams. The MAP-estimated PPM for "alo" is shown in Figure 2.1(4), where we see substantial movement of the senone timing distributions.

2.5 Detection-based KWS and ASR with PPM

Our proposed detection-based ASR architecture consists of four steps:

i. We build a PPM for each in-vocabulary (IV) unigram word.

ii. For each test utterance, run parallel word detectors for the whole vocabulary.

iii. Use the resulting independent word detections to build a confusion network or word lattice.

iv. Use standard techniques to process the confusion network or lattice for KWS indexing [45] and LVCSR decoding [46].

Below we describe our confusion network and lattice construction methodologies.

2.5.1 Confusion network construction

The standard confusion network is derived from a decoding lattice as a more compact representation with relaxed word sequence constraints [33]. It requires that the posterior probability for each arc in the lattice is estimated (by running forward-backward algorithm), and that the temporal partial order between arcs is derived via lattice topology. Since there are word identity, start time, duration, and posterior probability estimates (by a logistic regression applied to the likelihood ratio detection score of Eq. 2.6) associated with each PPM detection, we can naturally adapt the algorithm of [33] to build confusion networks based on PPM detections rather than decoding lattices. For each test utterance, we first sort the PPM detections of all the IV words according to their start time, and initialize each detection as an equivalence class (formed by word identity, start and end times). Second, we perform intra-word clustering

to merge the equivalence classes of the same word identity, and then perform inter-word clustering based on phonetic similarity, resulting in a complete alignment of competing detections as confusion bins.

2.5.2 PPM-based lattice generation

The duration of a PPM detection is hypothesized as the one that gives the maximum detection function value of Eq. 2.6, which may not be as accurate as that derived from the HMMs based on frame likelihood. Since the KWS scoring metrics can accommodate small time differences between the detections and the true references, such approximated duration from PPM is generally sufficient for the KWS task. However, the confusion network algorithm relies on strict temporal order between word components for clustering and inaccurate durations can lead to suboptimal results. Moreover, the confusion network algorithm requires word posterior estimates for each detection; the raw PPM detection score is a likelihood ratio and applying a global logistic regression for normalization is known to give suboptimal posterior estimates. Therefore, we propose a lattice construction algorithm for the PPM framework to accommodate the duration uncertainties and rely on word acoustic likelihood only, as described below.

First, for each PPM detection, we express its joint likelihood of acoustic observations $O_{t,t+T}$ and hypothesized duration as

$$p(O_{t,t+T}, T|\theta_w) = p(O_{t,t+T}|T, \theta_w)P(T|\theta_w) \tag{2.8}$$

where $p(O_{t,t+T}|T, \theta_w)$ is given by Eq. 2.3 and further by Eq. 2.4 with the event

set normalized in time, and $P(T|\theta_w)$ is a word-specific gamma distribution. Second, for an arbitrary region between two word detections, e.g. non-speech silence or noise, we employ a separate silence model of homogeneous Poisson process for the observed acoustic events in that region that takes the form

$$\begin{aligned} p(O_{t,t+T}, T|\theta_{sil}) &= p(O_{t,t+T}|T, \theta_{sil})P(T|\theta_{sil}) \\ &= \prod_{p \in P} (\mu_p)^{n_p} e^{-\mu_p T} P(T|\theta_{sil}) \end{aligned} \tag{2.9}$$

where p represents either context-independent monophone or context dependent senone in the event set $\mathcal{P}$, μ_p is the homogeneous Poisson rate parameter for each p under the silence model θ_{sil} with $P(T|\theta_{sil})$ modeled by a gamma distribution. Thus, we have approaches to compute acoustic likelihoods given any word hypothesis or an arbitrary region of acoustic observations.

Our strategy is to define "words-on-nodes" lattices, where each word detection becomes a node and the edges encode the temporal sequence of detections with directed arcs that can accommodate a sensible amount of temporal overlap. We define the construction process using the following notation. We denote the set of all the detections within a given utterance as $\mathcal{D} = \{w_i\}_{i=1}^{N}$, and sort $\mathcal{D}$ according to each detection's start time. For each word detection $w_i \in \mathcal{D}$ with index i in time, we define a node with acoustic likelihood given by Eq. 2.8, and $t_s(w_i)$ as its start time. We refer to all observed acoustic events that have arrived during the course of w_i as set $\rho(w_i)$, which is also the set of events used to give the local maximum value of Eq. 2.6.

The goal is to produce a directed acyclic graph, where $\phi(w_i)$ is the set of word detections (nodes) that w_i has an outgoing edge to, such that any word

in $\phi(w_i)$ can follow w_i in the output word sequence. We make each detection w_i (except the final node defined as the end of the utterance) connect to at least one another next node (in time) w_j ($j > i$), which we require by that: (i) w_j does not consume any acoustic event arrived during w_i, i.e., no intersection between $\rho(w_j)$ and $\rho(w_i)$, and (ii) the time gap between observations of w_i and w_j does not exceed a maximum allowable time gap δ (initialized as 1 sec) if possible. If we denote $t'_s(w_i)$ as time of the first phonetic event observed in time within w_i, and $t'_e(w_i)$ as time of its last observed event, then condition (i) becomes $t'_e(w_i) < t'_s(w_j)$, and condition (ii) becomes $(t_s(w_j) - t'_e(w_i)) < \delta$.

Also, if there are no acoustic events between time interval $(t'_e(w_i), t_s(w_j))$, we connect w_i to w_j with a free edge. If there is, we add a new node as w_{sil} of which the acoustic likelihood is computed by Eq. 2.9 on the acoustic events between interval $(t'_e(w_i), t_s(w_j))$ and the duration is given by $(t_s(w_j) - t'_e(w_i))$; further, we connect w_i to w_{sil} and connect w_{sil} to w_j.

In this approach, we can finally obtain a directed acyclic graph where each node is associated with its word identity, acoustic likelihood, start time and duration. The procedure described thus far is illustrated graphically by an example in Figure 2.2. By replying on the phonetic event timing information to determine the temporal order of the word sequence, we relax the accurate estimation of word start and end times but still enable an appropriate lattice construction, with the unidentified phonetic events accounted by optionally added silence nodes.

Finally, we convert the graph into a standard lattice with word and acoustic likelihood on each arc, which can be processed by standard FST-based

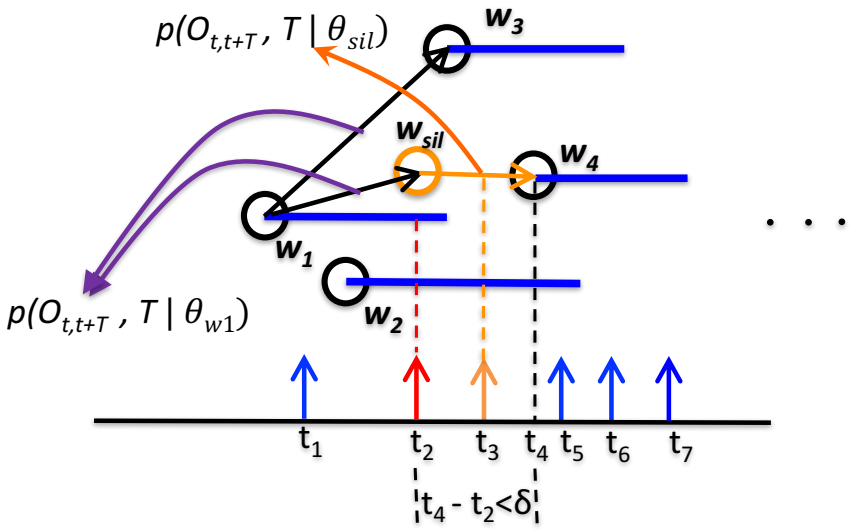


Figure 2.2: An illustration of how we build a words-on-nodes lattice, specifically in adding outgoing edges for detection w_1. First, the phonetic event at t_2 arrives during both w_1 and w_2, such that we do not connect node w_1 to w_2. Next, we find no intersection between $\rho(w_1)$ and $\rho(w_3)$, such that an edge is added between w_1 and w_3, and the acoustic likelihood on this edge is computed by Eq. 2.8 on the events at t_1 and t_2. Then we also find no intersection between $\rho(w_1)$ and $\rho(w_4)$, $t_s(w_4) - t'_e(w_1) = t_4 - t_2 < \delta$, and neither w_1 nor w_4 consumes event at t_3; thus, we add a new node w_{sil} at t_2, and the acoustic likelihood on the edge between w_1 and w_{sil} is given by Eq. 2.8 on the events at t_1 and t_2, and the acoustic likelihood on the edge between w_{sil} and w_4 is given by Eq. 2.9 on the event at t_3. Finally, $\phi(w_1) = \{w_3, w_{sil}\}$. We iterate this process for each detection w_i, $i = 1, \ldots, N$.

algorithms such as language model composition.

2.6 Experiments

In this section we describe our experimental setup and present the results of our evaluation.

2.6.1 Evaluation design

Incorporating the above developments, we perform a comprehensive keyword search evaluation in the IARPA Babel Program framework [10], which has released conversational telephone speech corpora for several languages. We measure our system performance on Haitian[4], Lao[5], Assamese[6], Bengali[7] and Zulu[8]. For each language there are two resource conditions: the full language pack (FullLP) contains approximately 80 hours of transcribed speech audio along with a pronunciation dictionary that covers all word types it contains; the limited language pack (LimitedLP) contains a 10 hour subset of FullLP. Language model text and pronunciation dictionary entries for LimitedLP are restricted to those that occur in the given 10 hours. In this chapter we only consider LimitedLP, which simulates low-resource conditions for a diverse set of languages. To evaluate system performance, we have a 10-hour development-testing search collection for each language while tuning on a 2-hour subset. Keyword sets are the official development lists generated by Babel participants for use before the evaluation period, which consist of approximately 2000 multi-word queries for each language.

We use two KWS scoring metrics, Actual Term-Weighted Value (ATWV) and Oracular Term-Weighted Value (OTWV) as described below. ATWV was developed for the NIST 2006 STD evaluation [8] and is the primary metric in the Babel program. First, for each hypothesized keyword detection, i.e.

[4]Language collection release `IARPA-babel201b-v0.2b`.
[5]Language collection release `IARPA-babel203b-v3.1a`.
[6]Language collection release `IARPA-babel102b-v0.5a`.
[7]Language collection release `IARPA-babel103b-v0.4b`.
[8]Language collection release `IARPA-babel206b-v0.1e`.

each putative hit, the KWS system is required to report its begin and end time, and a posterior score indicating how likely the putative hit is a true keyword occurrence. Then a Term-Weighted Value (TWV) is defined as:

$$TWV(\theta) = 1 - \frac{1}{K}\sum_{w=1}^{K}\left(\frac{N_{\text{Miss}}(w,\theta)}{N_{\text{True}}(w,\theta)} + \beta\frac{N_{\text{FA}}(w,\theta)}{T - N_{\text{True}}(w,\theta)}\right) \tag{2.10}$$

where K is the total number of keywords, and θ is the detection threshold (i.e. only keyword detections with posteriors over θ are considered in scoring and otherwise are ignored); then given θ, $N_{\text{Miss}}(w)$ is the number of missed detections of keyword w, $N_{\text{FA}}(w)$ the number of false alarms of w, $N_{\text{True}}(w)$ the number of reference occurrences of w, and β is a constant (specified as 999.9). The range of $TWV(\theta)$ is $(-\infty, 1]$. ATWV requires scores to be both normalized across keyword such that a single global threshold θ can be set, as well as well calibrated against the true posterior probability of correctness such that the global threshold θ is 0.5.

Second, Oracular Term-Weighted Value (OTWV) is defined assuming the keyword-specific optimal threshold $\hat{\theta}_w$ is used instead of 0.5. Specifically, for each keyword query, we can choose the detection threshold $\hat{\theta}_w$ that maximizes the keyword-specific $TWV(w, \theta_w)$:

$$\begin{aligned}\hat{\theta}_w &= \operatorname*{argmax}_{\theta_w} TWV(w,\theta_w) \\ &= \operatorname*{argmax}_{\theta_w} 1 - \left(\frac{N_{\text{Miss}}(w,\theta_w)}{N_{\text{True}}(w,\theta_w)} + \beta\frac{N_{\text{FA}}(w,\theta_w)}{T - N_{\text{True}}(w,\theta_w)}\right)\end{aligned} \tag{2.11}$$

Thus, OTWV over the complete keyword set is

$$OTWV = \frac{1}{K}\sum_{w=1}^{K} TWV(w,\hat{\theta}_w) \tag{2.12}$$

Since OTWV does not require scores to be normalized across keyword, it is a measure only of ranked list quality. OTWV is also an upper bound on a system's ATWV. The NIST F4DE scoring tool is used for reference alignment, and YES/NO decisions are made based on posterior scores.

In addition, we may decompose search performance into in-vocabulary and out-of-vocabulary keyword sets.

2.6.2 System implementation details

The DNN infrastructure of the Kaldi toolkit [47] is used as the input phonetic acoustic model. Here, we first train a standard GMM-based, speaker adaptively trained model to obtain HMM-state alignments and fMLLR feature transforms. Next, we train a 5-layer DNN of p-norm units with $p = 2$ [40]. The basic input features are 13-dimensional PLP augmented with 3-dimensional pitch and POV features, and spliced by 3 frames; then the 48-dimensional feature is reduced to 40 dimensions using linear discriminant analysis (LDA). Adaptation with maximum likelihood linear transforms with semi-tied covariance (MLLT/STC) and fMLLR is applied, and 9-frame context windows are stacked to represent the center frame. Thus, the resulting inputs to the DNN are 360 dimensions, and the outputs are posteriors over context dependent HMM-states where the number and identity depend on the language. The context-independent PPM framework operates on monophone posteriorgrams, which are then derived by summing posterior dimensions corresponding to the same center phone.

To obtain pronunciations for OOV keywords, we use the Sequitur G2P

toolkit [41], a data-driven G2P converter based on joint-sequence models. We use each language's LimitedLP lexicon with pairwise examples of word and pronunciations to train a G2P model, and use the trained model to generate the pronunciation for a given OOV keyword. Each dictionary-based PPM is synthesized according to the prescription given in [23], and updated by MAP estimation if training exemplars are available. For multi-word keywords, we construct the dictionary-based PPM for each unigram in the multi-word keyword, update each unigram PPM if exemplars for that unigram are available, and then concatenate unigram PPM into a multi-word PPM, as described in [24].

For OTWV calculation, we can use the PPM likelihood ratio detection function directly without tuning any score normalization parameters. However, for the ATWV calculation we must provide confidence scores normalized across keywords. Following [24], we use a simple two-parameter logistic regression (slope and bias) to map PPM detection function scores to posterior probability estimates and apply the term-specific thresholding technique described in [44]. Following [45], we estimate these logistic regression parameters using a 2 hour subset of the 10 hour development set we use for testing. Separately, we performed cross-validation experiments to confirm that this minor train-on-test violation did not unfairly impact our results.

Our KWS baseline is the Kaldi LVCSR-based keyword search system [45], which is outfitted with the identical DNN acoustic model we use for the PPM. OOV performance is compared against the proxy keyword search [42], which derives putative hits from LVCSR word lattices.

2.6.3 Results with context-independent PPM

Table 2.1 shows the LimitedLP KWS results on the five languages using the Kaldi LVCSR and PPM systems, as well as the combination of the two. Also listed are the relative fusion gains over the baseline, as well as average performance values over the five languages. Consistent with the results in [24], we find that LVCSR-based search dominates ATWV, with the PPM achieving on average only 42% of the baseline performance. However, we find that PPM search gives much more competitive results on OTWV performance, a metric that evaluates the quality of the ranked list *independent of* the consistency of confidence scores across keywords. This OTWV-ATWV divergence is a consequence of the PPM's suboptimal score normalization, which is performed using a simple logistic regression applied to the likelihood ratio detection score of Eq. 2.6. Indeed, the LVCSR search system computes true lattice posterior scores, which normalize each lattice arc likelihood by all the other words that might have accounted for the same acoustic observations. This is a much more powerful normalization scheme, but it does come at the larger computational cost of decoding the whole vocabulary at indexing time. For keyword applications that do not require score normalization, the PPM system provides on average 66% of LVCSR baseline OTWV performance with a much smaller index processing time and size (see [24] for details).

If we consider OOV keyword search ATWV in isolation, we can see that the dictionary-based PPM achieves comparable results with the state-of-the-art WFST-based proxy keyword search. The PPM outperforms on Haitian and Zulu, while falling short on Lao, Assamese and Bengali, so it interesting to

Table 2.1: LVCSR, PPM, and combined search performance for the five languages, along with relative gain from combination over the LVCSR baseline alone. Averages are over the corresponding individual language fields.

Language	System	OTWV (All)	ATWV (All)	ATWV (IV)	ATWV (OOV)
Haitian	LVCSR	0.54	0.44	0.49	0.23
	PPM	0.36	0.21	0.20	0.25
	Comb	0.60	0.48	0.51	0.35
	% Gain	11.1	9.1	4.0	52.2
Lao	LVCSR	0.51	0.41	0.43	0.22
	PPM	0.32	0.16	0.17	0.12
	Comb	0.57	0.44	0.47	0.26
	% Gain	11.8	7.3	9.3	18.2
Zulu	LVCSR	0.28	0.17	0.30	0.09
	PPM	0.27	0.11	0.06	0.14
	Comb	0.41	0.24	0.32	0.19
	% Gain	46.4	41.2	6.7	111.1
Assamese	LVCSR	0.37	0.25	0.31	0.10
	PPM	0.21	0.08	0.08	0.07
	Comb	0.42	0.28	0.34	0.14
	% Gain	13.5	12.0	9.7	40.0
Bengali	LVCSR	0.38	0.27	0.35	0.13
	PPM	0.22	0.10	0.10	0.09
	Comb	0.43	0.30	0.37	0.17
	% Gain	13.2	11.1	5.7	30.8
Averages	LVCSR	0.42	0.31	0.38	0.15
	PPM	0.28	0.13	0.12	0.14
	Comb	0.49	0.35	0.40	0.22
	% Gain	19.2	16.1	7.1	50.5

consider what language-specific properties may be driving this variation. For Zulu, an agglutinative language with a unusually high keyword OOV rate, the PPM system achieves much closer overall KWS performance with LVCSR, indicating PPM's advantage for truly low-resource settings with woefully incomplete pronunciation dictionaries. Note that the PPM usually gives

comparable or even higher OOV ATWV results than IV, since we find that PPM search is more sensitive to keyword length and OOV keywords tend to be longer.

Given the distinct lexical modeling strategies employed in the LVCSR baseline and PPM search systems, as well as the substantial relative performance variation across language, some degree of complementarity is to be expected. Even though the PPM overall performance substantially trails the LVCSR baseline on all five languages, we measured a 16% average relative improvement of both ATWV and OTWV in combination. Moreover, the comparable performance of PPMs and proxy keyword search for OOVs combine to produce an average ATWV relative increase of 50% over proxies alone. While in-vocabulary PPM performance lags LVCSR the most, we still post an average relative gain of 7% in fusion.

In terms of runtime comparison between proxy keyword search and PPM OOV search on the 10 hour development set, we compare the average runtime of five languages for the three stages of operation, in terms of CPU time (in seconds). First, for indexing time on the 10 hour search collection, proxy keyword search takes 5,736 seconds to make an inverted index from decoding lattices, while the PPM system takes 256 seconds to extract phonetic events from monophone posteriorgrams. Second, for model construction, it takes 2.4 seconds to generate word proxies for each keyword, while it takes 0.01 seconds to construct one dictionary prior PPM. Finally, for searching the index, proxy search takes 0.55 seconds for each keyword, while the PPM search takes 0.08 seconds (computed using the benchmark information provided in [24]).

In all three categories, we find that OOV search with PPMs is significantly more efficient in time than proxy keyword search. It does require an additional phone event index, but as demonstrated in [24], the index construction time and size are negligible.

2.6.4 Results with context-dependent PPM

We evaluate the efficacy of incorporating context-dependency into the original context-independent PPM framework (without lattice construction), where the word posterior is approximated by a logistic regression applied to detection score of Eq. 2.6. The results of two languages[9] are shown in Table 2.2. We see that context-dependent PPM on senone events significantly outperforms the monophone baseline in nearly all categories, but remains the same for multi-word keywords. We can account for this by the fact that more monophone events are observed in the generally longer multiword queries, which limits the additional benefit of more detailed triphone patterns.

Finally, it is important to note that even though the senone set (approximately 2000 units) is much larger than monophone set ($\sim$50 dimension), in practice the PPM search index size is on average only 2.2 times larger than before. This is a result of the fact that the increase in posteriorgram units does not substantially reduce event sparsity since the new units are generally mutually exclusive. It follows that the PPM's storage advantages highlighted in [24] are maintained despite the increased model detail.

[9] From Table 2.1 we observe that the PPMs perform similarly on Haitian and Lao, and also similarly on Assamese and Bengali, so that we only choose one from each language pair for subsequent evaluations; also, the unusually high OOV rate makes Zulu a challenging dataset and we leave it to future work.

Table 2.2: PPM search performance for Haitian and Bengali, along with relative gain from using senone over monophone events.

Language	PPM System	OTWV (All)	ATWV (All)	ATWV (IV unigram)	ATWV (unigram)	ATWV (multiword)
Haitian	# keywords	1921	1921	418	573	1348
	monophone	0.361	0.212	0.119	0.127	0.249
	senone	0.380	0.225	0.158	0.159	0.253
	% Gain	5.3	6.1	32.8	25.2	1.6
Bengali	# keywords	1967	1967	603	926	1041
	monophone	0.222	0.101	0.029	0.041	0.154
	senone	0.237	0.111	0.061	0.061	0.155
	% Gain	6.8	9.9	110.3	48.8	0.6

2.6.5 Results with PPM-based lattice generation

We refer to the independent keyword-specific PPM search evaluated above (without lattice construction) as the baseline in Table 2.3, and compare with the PPM's confusion network and lattice based KWS. Since keywords tend to have lower unigram probabilities in training transcript, to increase the keyword recall we keep more detections for words that occur rarely during training. To accomplish this we prune PPM detections of each IV unigram based on its unigram probability using empirically determined thresholds (i.e. tuned on the 2 hour subset of the 10 hour development-testing set as discussed in Section 2.6.2). Further, confusion networks and lattices are obtained as described in Section 2.5, and we compose them with a FST-based language model to give each arc a trigram language model prior, with a tuned acoustic scaling factor.

Table 2.3 shows that the adapted confusion network approach does not outperform baselines, a result of suboptimal duration and posterior estimation

Table 2.3: KWS performance (IV unigrams) comparisons between keyword-specific PPM search and lattice-based approach.

Language	PPM System	OTWV	ATWV
Haitian	baseline, monophone	0.241	0.119
	confusion network, monophone	0.233	0.066
	lattice, monophone	0.257	0.129
	% Gain	6.6	8.4
	baseline, senone	0.298	0.158
	lattice, senone	0.305	0.175
	% Gain	2.3	10.8
Bengali	baseline, monophone	0.113	0.029
	lattice, monophone	0.122	0.029
	% Gain	8.0	0.0
	baseline, senone	0.162	0.061
	lattice, senone	0.173	0.080
	% Gain	6.8	31.1

Table 2.4: WER performance from PPM and HMM lattices.

Language	System	WER
Haitian	PPM lattice, monophone	74.1
	PPM lattice, senone	69.8
	HMM, senone	59.6
Bengali	PPM lattice, monophone	80.5
	PPM lattice, senone	77.9
	HMM, senone	66.8

issues discussed in Section 2.5. The proposed words-on-nodes lattice generation algorithm, which incorporates the competing hypotheses and contextual constraints into the PPM search, leads to consistent KWS improvements for both monophone and senone event-based systems. We also find that, combining context-dependency and PPM lattice generation yields significant gains over the original monophone baseline.

Finally, Table 2.4 shows the lattices generated by PPM framework can also

provide reasonable ASR performance. Though its WER trails the DNN-HMM systems, it has obvious computational merit. The PPM index is created about 2x faster than real time (RT), and each IV word can be detected in parallel with speeds 500,000x faster than RT [26]. The subsequent PPM lattice construction complexity is of order $O(N^2)$, where N is the number of detections in an utterance; since we only consider connecting each detection to its close neighbors (within a maximum allowable time gap δ like 1 sec in Section 2.5.2), the runtime in practice is in excess of 1,000x faster than RT. Thus, we find the overall runtime of PPM decoding and lattice generation much more efficient than the real-time factor 8.41 of the DNN-HMM based lattice generation (comparing based on one single core of a 2.40-GHz Intel Xeon processor). The subsequent operations of language model composition and lattice indexing are efficiently implemented in a WFST-based framework as before [45].

2.7 Conclusion

In this chapter we have demonstrated that the point process model framework provides a viable keyword search platform for low-resource settings. It is highly complementary with state-of-the-art LVCSR techniques, posting substantial fusion gains for every language evaluated. On its own, it provides state-of-the-art handling of OOV keywords, but also produces dramatics gains when combined with proxy keyword search outputs. The incorporation of context-dependent phonetic events into the PPM framework produces substantial further improvements with only a small increase in computational complexity.

Finally, as evidenced by comparatively large gaps between ATWVs and OTWVs, the substandard score normalization achievable with PPMs remains a major challenge. Therefore, we have introduced a lattice generation algorithm specifically tailored to the PPM setting, and demonstrated that KWS via PPM lattice generation produces further performance improvements by incorporating language models and better score normalization. Furthermore, lattices from PPM support LVCSR decoding, which give reasonable performance for a first attempt on a difficult task.

Chapter 3

Spoken Document Classification with ASR

To discover what we are looking for from vast audio collections, the development of new computational tools is required to help analyze, organize and search these extensive amounts of information. Spoken document classification is one such human language technology that determines which class(es), if any, each of a set of documents belongs to. The classes, also called categories or labels, can be predefined based on themes, sentiments, or any other attributes. Most retrieval systems today contain multiple components that use some form of classifier [4], such as:

- In Topic Detection and Tracking [6], each incoming news story needs to be classified as to whether or not it discusses a previously known topic.
- Given large recording collections of academic lectures, a lecture browser system can be built to allow users to type a query, search through lectures and receive the relevant portions [48]. Lectures can be classified into different topic categories, such that queries can be constrained by

allowing users to specify a topic category before searching.

- In automatically measuring customer satisfaction for phone calls in a contact center, recorded conversations can be classified into distinct points, such that dissatisfied customers and areas for service quality enhancement can be identified [49].

These examples show the general importance of classification in speech retrieval applications. In this chapter[1], we examine spoken document classification via automatic speech recognition (ASR) transcriptions.

3.1 Introduction

Since audio data lacks the paragraphs and punctuation markings that naturally define semantically coherent chunks of text, long audio recordings of varying label/topic shifts are usually first segmented according to some task specific criteria, manually or by an automatic segmentation system [6, 7]. Then the standard approach to spoken document classification is to

i. develop ASR systems to decode each speech segment into word sequences,

ii. produce intermediate vector representations of the hypothesized word sequences for each segment, and

iii. learn a classifier from text/label pairs and apply it to the vector representation of each segment independently.

[1] Large portions of this chapter have been published in [16, 50].

However, such standard approach has many drawbacks, especially in a low-resource scenario: building ASR and document classifiers in a new language requires a large amount of transcribed speech and class-labeled texts in the language, neither of which may be present. Furthermore, accurate topic inference or language understanding in general may require interpretation from adjacent segments. For instance, tasks such as anaphora resolution or entity disambiguation critically depend on contextual clues.

To study these challenges, we evaluate our spoken document classification performance in the DARPA LORELEI (Low Resource Languages for Emergent Incidents) Program framework. The program's goal is to develop human language technologies to support humanitarian assistance and disaster relief operations in locations where a low-resource language is spoken, also referred to as an incident language (IL) in the LORELEI terminology [14, 51]. To provide situational awareness via IL sources, one component task in LORELEI, called the Situation Frame (SF) task, involves building systems to provide meta-data for text and speech documents. These documents and associated meta-data are collectively referred to as situation frames (SFs) and consist of the following items:

- situation type, also simply referred to as topic,
- geographic localization,
- status (temporal, resolution or urgency) of the situation.

An SF system is required to automatically identify all the SFs covered in the text or speech collection in the IL. In this chapter, we focus on building topic

identification (topic ID) technology to enable situation type identification from speech. Thus, we consider topics as the classes in the document classification definition through this chapter.

In order to simulate realistic disaster scenarios, the LORELEI speech corpora are divided into IL corpora – corpora which typically contain unlabeled data in a low-resource language pertaining to one or more emergent disasters – and related language corpora for which annotated data, possibly from high-resource languages, is provided. In both cases the audio data is collected "in the wild", and for a diverse set of languages. These data are collected, manually segmented, and annotated by APPEN [52] for the LORELEI program. We refer to each unsegmented audio file as one spoken document. Since audio file segmentations are provided, each document consists of a sequence of segments, and each segment lasts around one minute on average and no more than 2 minutes. There are 11 predefined topics chosen according to the

Table 3.1: Topic labels defined in the LORELEI Speech SF task.

Topic scope	Topic label (Situation Type)
In-domain	Evacuation
	Food Supply
	Urgent Rescue
	Utilities, Energy, or Sanitation
	Infrastructure
	Medical Assistance
	Shelter
	Water Supply
	Civil Unrest or Wide-spread Crime
	Elections and Politics
	Terrorism or other Extreme Violence
Out-of-domain	Out-of-domain

LORELEI program scope, as shown in Table 3.1. Any speech segment categorized by at least one of these topics is defined as in-domain data, otherwise as out-of-domain that can be viewed as the 12th topic label. Table 3.2 shows an example spoken document that is split into 7 segments with varying topic.

In this chapter, we focus particularly on the IL scenario for which the only annotated data are from related (development) languages in addition to a very small amount of IL topic labeled data or IL transcribed speech (minutes rather than hours) which may be obtained.

Table 3.2: An example of a single spoken document that consists of seven spoken segments in the LORELEI US English corpus.

Doc ID	Segment ID	Sampled sentences	Topic
080	080_001	turning to Tennessee where eleven people have now died in historic wildfires ... hundreds of buildings have been torched ...	Shelter
080	080_002	yeah you have a number of people missing but we don't know the exact number ...	Out-of-domain
080	080_003	... and he said that the search and rescue effort yesterday ended and now today it is search and recovery ...	Urgent Rescue
080	080_004	... so many homes damaged destroyed ...	Shelter
080	080_005	... just looking at the devastation now . because we saw a few homes and you know a few cars, it is really bad ...	Shelter
080	080_006	... but people in town it sounds like now are questioning how fast they were notified to get out ...	Evacuation
080	080_007	... since they were forced to evacuate so a lot of them will be seeing their homes and properties for the first time tonight ...	Evacuation, Shelter

3.2 Related work

Prior work of topic ID on speech [53, 54, 55, 56] has focused on conversational telephone speech such as LDC's Fisher and Switchboard collections, where topic ID was performed for each whole conversation. Since the two participants of each conversation were prompted to speak on one single topic, no conversation segmentation was needed. Furthermore, since each conversation contains a single topic and lasts 5-10 minutes, the classification task is relatively straightforward. Document representations are bag-of-words multinomial representations over word or phone n-grams, with or without dimensionality reduction like Latent Semantic Analysis (LSA) [57] or Latent Dirichlet allocation [58]. Topic classifiers are focused on linear classifiers, such as Naïve Bayes, logistic regression, support vector machine (SVM), etc.

Extensive work on text classification has been explored to date, where each text data instance can be a sentence or a document. For example, word sequences can be mapped to word embedding vectors and used as inputs to convolutional neural network with a final softmax classification layer [59, 60, 61]. [62] introduces using recursive neural network and [63] applies recurrent neural network. Furthermore, [64] examines producing sentence representations by an attention mechanism that learns attention weight distributions over words, and [65] proposes to use a hierarchical attention network to learn both word- and sentence-level attentions. Attention mechanisms demonstrate efficacy in improving classification performance, through enabling the models to attend differentially to more and less important contexts [65].

However, all the above work is focused on performing single-label classification for each data instance (i.e. each single sentence, conversation, or document) individually, and independently from the rest of data instances. Data instances in close proximity to each other may incorporate contextual information that can be exploited by contextual modeling.

The LORELEI collections provide a challenging and realistic scenario, where wildly collected audio recordings can be extremely long, of varying length, and contain multiple topic shifts at variable locations in the audio. For this reason each audio document in the LORELEI data is first segmented by APPEN [52], and then topic classification is required on the much shorter resulting segments. To solve the LORELEI task, prior work [66] used a mismatched ASR to directly decode IL speech, while [16] proposed sharing common phonemic representation among languages and transferring acoustic models trained on higher-resource (potentially related) language(s). After ASR, [66, 16] translated both development (dev) and incident languages into English words, used the translated dev language data along with the given topic label annotations to learn English-language topic models and then classify the translated IL data.

Instead of using ASR to convert speech into sequences of words, [67, 16] also investigated unsupervised techniques to automatically discover and tokenize IL speech segments into phone-like units via acoustic unit discovery (AUD) or word-like units via unsupervised term discovery (UTD). However, only small amount of IL topic labels might be available to learn classifiers based on AUD/UTD tokenized segments, though [16] showed marginal gains

by combining them with the above cascade approach that implemented ASR, machine translation (MT) and operated on English words.

However, in all the approaches above, topic ID was performed on each speech segment individually, without exploring the contextual information between adjacent segments. We also note that our topic ID task, which is formulated as multi-label classification for each speech segment in a spoken document, is similar to the domain or intent classification in a multi-turn spoken language understanding (SLU) component of a dialog system [68, 69, 70]. One conversation session between user and dialog system, which can be viewed as one spoken document, may include multiple turns, and the user query in each turn is a spoken segment; thus, each segment needs to be classified into one of the supported domains or user intents, as classified into topic(s). [68, 69, 70] have shown that SLU may require contextual interpretation from the dialog history, and the SLU models that incorporate the semantic contexts of preceding user utterances and system outputs outperform those without context. Therefore, in this chapter, we also investigate if the propagation of contextual information across spoken segments can improve topic ID, although the spoken segment that is one minute long on average in our case is often much longer and more semantically self-contained than the typical utterance of a few words in SLU systems.

3.3 Universal phone set ASR

This section examines how to build an ASR for an incident language where little or no transcribed speech data is available and pronunciation lexicon is

severely incomplete. Previous approach [71] has explored cross-language ASR transfer assuming shared phonemic representations, using the multilingual GlobalPhone corpus [72] and manual phone mapping based on the IPA (International Phonetic Alphabet) scheme. [12] further uses a set of languages sharing the X-SAMPA phone sets [73] from Babel corpus [10].

Our approach is similar to [12]. We attempt to provide language universal acoustic models by training on many languages sharing a common phonemic representation. We then transfer these models to a new language via a pronunciation lexicon with the same phonemic representation as used in training. We refer to this ASR as universal phone set ASR. We also use a selection of BABEL languages for training. Diphthongs and triphtongs are split into their constituent phones to reduce the number, and enforce sharing, of phonemes. Also, as in [12], we standardize the representation of tone (tonal trajectory) across all training languages. The final acoustic models are time-delay neural networks (TDNNs, [74]) trained with the lattice-free version of the maximum mutual information (LF-MMI) criterion [75].

During a LORELEI evaluation we may also have access to a few hours (2-10) of consultation with a native informant (NI), a native speaker of the IL. From these interactions we can collect an additional 15-30 minutes of IL speech transcriptions. We use this data to adapt the ASR for both languages using the same weight transfer approach as in [76]. Since the source languages and ILs use the same phoneset, all layers of the seed neural network (trained on the source languages), including the final layer, are transferred and trained for one epoch on the IL transcribed data.

3.4 Document representation and classification

To leverage the supervised topic annotations of speech segments in multiple dev languages, we represent each speech segment in all languages as a bag of English words. We derive this representation by building ASR systems to decode the speech and then translate each decoded word into its most likely English translations. We propose to use the probabilistic bilingual translation tables employed in the MT systems, i.e. bilingual lexicons, rather than full-blown MT systems to relax the dependency on fully developed IL-to-English MT pipeline that could be unavailable for very-low-resource languages.

SVM or neural network (NN) based topic classifiers can then be learned by using these English word representations of speech segments in foreign languages along with their associated topic labels. Thus, using only a translation lexicon, we can always perform topic ID on an IL without its transcribed or topic-labeled speech by using the unadapted universal phone set ASR to decode and translate its speech segments into English words.

3.4.1 Learning spoken segment representations

Since English word sequences generated using translation tables lack proper syntax, we represent speech segments using a bag-of-words model over the generated English words. Each speech segment is represented by a vector of unigram occurrence counts over the generated English word sequences and scaled to produce a term frequency-inverse document frequency (tf-idf) feature, which is then normalized to ℓ_2 norm unit length.

Latent Semantic Analysis (LSA) [57] transformation can then be learned

from the tf-idf features. This transformation effectively merges the dimensions corresponding to words with similar meanings, and maps the high-dimensional tf-idf vectors to a much smaller dimension vector space.

We can also append other auxiliary features to the tf-idf or LSA representations of speech segments. Since our datasets contain segments with music, many of which are out-of-domain, we found that features indicating the substantial presence of music are particularly useful. To generate these features, we build music detectors from the MUSAN dataset [77] and for each speech segment the music detector produces a posterior probability that a substantial portion of music is present. Denoting the tf-idf/LSA vector as $\mathbf{x} \in \mathbb{R}^d$, the music posterior as $\delta \in (0, 1)$, and the vector concatenation operation as $\oplus$, our new representation can be created as $\mathbf{x} \oplus \delta$.

3.4.2 Non-contextual modeling using SVM and NN

Since each speech segment is represented by a vector $\mathbf{x}$ and can be associated with one or multiple topics, we perform topic ID by doing multi-label classifications. The baseline approach is the binary relevance method, which independently trains one binary SVM classifier for each label, and a segment is evaluated by each classifier to determine if the respective label applies. We use stochastic gradient descent (SGD) based linear SVMs with hinge loss and ℓ_2 norm regularization [78, 79].

Another approach based on feedforward NN[2] is to use an output layer with sigmoid output nodes, one for each label, and train the NN to minimize

[2]We simply use NN to refer to the multi-layer perceptron in the following sections.

the binary cross entropy loss defined as

$$\mathcal{L}(\Theta_{\text{nn}}; \mathbf{x}, \mathbf{y}) = -\sum_{k=1}^{K} (y_k \log o_k + (1 - y_k) \log(1 - o_k)) \tag{3.1}$$

where Θ_{nn} denotes the NN parameters, $\mathbf{y}$ is the target binary vector of topic labels, o_k and y_k are the output and the target for label k, and the number of unique labels $K = 12$.

3.4.3 Contextual modeling using RNN

We explore using recurrent neural network (RNN) to capture the dependencies between context segments. Different RNN variants can be used such as the Elman RNN, long short-term memory (LSTM), or gated recurrent unit (GRU). We denote an RNN simply as a mapping $\phi : \mathbb{R}^d \times \mathbb{R}^{d'} \rightarrow \mathbb{R}^{d'}$ that takes a d dimensional input vector $\mathbf{x}$ and a d' dimensional state vector $\mathbf{h}$ and outputs a new d' dimensional state vector $\mathbf{h}' = \phi(\mathbf{x}, \mathbf{h})$.

Consider a spoken document that consists of n spoken segments, as exemplified in Table 3.2. For each $i = 1 \ldots n$, the segment i is represented by a vector $\mathbf{x}_i \in \mathbb{R}^d$. The document is represented as $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_n]$. We encode $\mathbf{X}$ using a bidirectional RNN (BiRNN), and the model parameters Θ_{rnn} associated with this BiRNN layer are ϕ_f, $\phi_b : \mathbb{R}^d \times \mathbb{R}^{d'} \rightarrow \mathbb{R}^{d'}$. Thus the segment representation vectors are encoded by forward and backward RNNs as

$$\begin{aligned} \mathbf{f}_j &= \phi_f(\mathbf{x}_j, \mathbf{f}_{j-1}) \quad \forall j = 1 \ldots n \\ \mathbf{b}_j &= \phi_b(\mathbf{x}_j, \mathbf{b}_{j+1}) \quad \forall j = n \ldots 1 \end{aligned} \tag{3.2}$$

We assume zero initial state vectors $\mathbf{f}_0$ and $\mathbf{b}_{n+1}$. And a contextual representation is induced as

$$\mathbf{h}_i = \mathbf{f}_i \oplus \mathbf{b}_i \quad \forall i = 1 \dots n.$$

We denote the entire operation as a mapping $\text{BiRNN}_{\Theta_{\text{rnn}}}$:

$$(\mathbf{h}_1 \dots \mathbf{h}_n) \leftarrow \text{BiRNN}_{\Theta_{\text{rnn}}}(\mathbf{x}_1 \dots \mathbf{x}_n).$$

Therefore, instead of the non-contextual $\mathbf{x}_i$, the contextual $\mathbf{h}_i$ is used as input to the feedforward fully connected layers for final classification:

$$\mathbf{o}_i \leftarrow \text{NN}_{\Theta_{\text{nn}}}(\mathbf{h}_i) \quad \forall i = 1 \dots n$$

where $\mathbf{o}_i$ denotes the final output vector. The joint loss

$$\mathcal{L}(\Theta_{\text{rnn}}, \Theta_{\text{nn}}) = \sum_{i=1}^{n} \mathcal{L}(\Theta_{\text{nn}}; \mathbf{h}_i, \mathbf{y}_i)$$

is calculated by Eq. 3.1.

3.4.4 Contextual modeling using attention

Consider a spoken document $\mathbf{X}$ as above. For each target segment $\mathbf{x}_i$, RNNs implicitly encode its context segments as $\mathbf{f}_{i-1}/\mathbf{b}_{i+1}$, but the RNN non-linear transformations make it hard to control the interaction between segments. Instead, we explicitly perform a convex combination of the target and context segments using an attention mechanism [80].

For each $i = 1 \dots n$, now consider classifying $\mathbf{x}_i$. We aim to produce a new contextual vector representation $\mathbf{c}_i$ to replace $\mathbf{x}_i$, by combining $\mathbf{x}_i$ and its contexts $\mathbf{X} \setminus \mathbf{x}_i$. Then each $\mathbf{c}_i$ is followed by fully connected layers for final classification as in Section 3.4.2. To do so, let z_i be a categorical latent variable with sample space $\{1 \dots n\}$, which encodes the desired selection among $\mathbf{X}$

based on a query $\mathbf{q}_i$. We let the query be $\mathbf{x}_i$ itself, i.e., $\mathbf{q}_i = \mathbf{x}_i$, since $\mathbf{x}_i$ has been produced specifically to encode the semantic information pertaining to segment i. Then we assume the source position to be selected and attended to follows a distribution, $z_i \sim p(z_i = j|\mathbf{q}_i, \mathbf{X}), \forall j = 1 \dots n$, and therefore the contextual representation $\mathbf{c}_i$ is defined as an expectation:

$$\begin{aligned} \mathbf{c}_i &= \mathbb{E}_{z_i \sim p(z_i|\mathbf{X}, \mathbf{q}_i)}[\mathbf{x}_{z_i}] = \sum_{j=1}^{n} p(z_i = j|\mathbf{q}_i, \mathbf{X})\ \mathbf{x}_j \\ &= \sum_{j=1}^{n} \alpha_{ij} \mathbf{x}_j \end{aligned} \tag{3.3}$$

The weight α_{ij} of each $\mathbf{x}_j$ is computed by

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})}, \quad \forall j = 1 \dots n \tag{3.4}$$

where $e_{ij} = f(\mathbf{q}_i, \mathbf{x}_j)$, called an alignment model [80] that scores how important the segment j is to help classify the query segment i. We parameterize it with a single-layer NN,

$$\begin{aligned} e_{ij} &= \mathbf{w}^T \sigma(\mathbf{W}_1 \mathbf{q}_i + \mathbf{W}_2 \mathbf{x}_j + \mathbf{b}_1) + b_2 \\ &= \mathbf{w}^T \sigma(\mathbf{W}_1 \mathbf{x}_i + \mathbf{W}_2 \mathbf{x}_j + \mathbf{b}_1) + b_2, \quad \forall j = 1 \dots n \end{aligned} \tag{3.5}$$

where σ is an activation function, and $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d' \times d}, \mathbf{w}, \mathbf{b}_1 \in \mathbb{R}^{d'}, b_2 \in \mathbb{R}^1$ are the weight matrices and jointly learned with all the other NN parameters. Note that to classify the target $\mathbf{x}_i$, the contexts close to $\mathbf{x}_i$ can be more relevant than the distant ones, so we can also use a truncated context window and only consider its L/R nearest left/right contexts, i.e., for each $j = \max(0, i - L) \dots \min(i + R, n)$ in Eq. 3.3, 3.4 and 3.5. The complete modeling framework

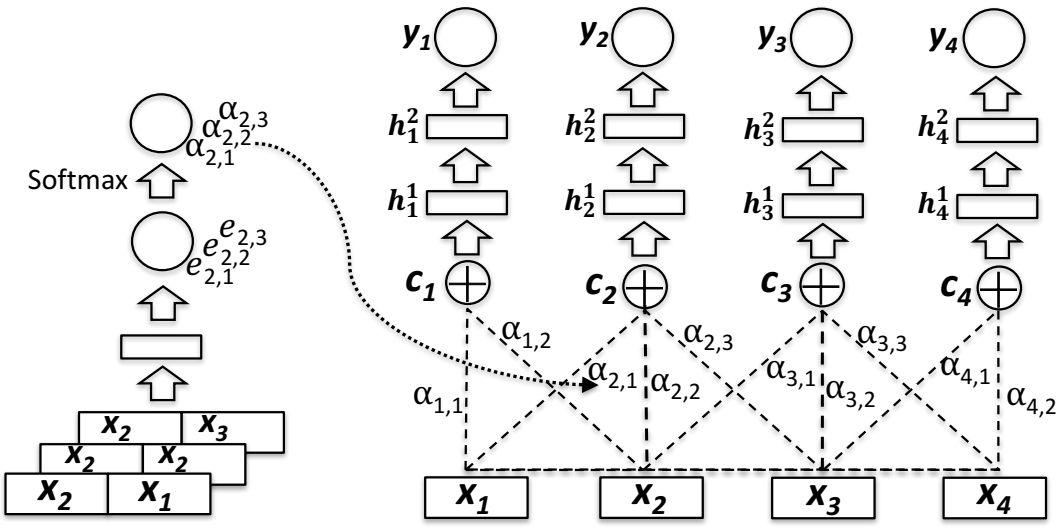


Figure 3.1: Illustration of the proposed contextual modeling using attention, which operates on a spoken document of 4 segments, and leverages each 1-nearest left and right context segments to classify the target $\mathbf{x}_i$, for each $i = 1 \dots 4$.

is illustrated in Figure 3.1, which uses the 1-nearest left and right contexts (i.e. when $L = R = 1$).

The intuition behind such process is that, although the overall feature vector x_i may not be salient enough to produce high posteriors for the correct topic labels, certain feature dimensions in x_i are indicative of the correct topics, so that the alignment model of Eq. 3.5 can still capture those informative feature dimensions and give the useful context segments higher scores e_{ij} and higher weights α_{ij}. The weights are used in a convex combination of Eq. 3.3 such that the useful context features are explicitly combined to produce a contextual representation $\mathbf{c}_i$.

In contrast with the deterministic RNN mapping, the attention mechanism allows for selectively using the contexts in a dynamic manner. Consider that, given the left contexts of $\mathbf{x}_i$, the forward RNN produces a context vector $\mathbf{f}_{i-1}$

as in Eq. 3.2, and the context vector $\mathbf{f}_{i-1}$ is used in a deterministic function $\phi_f(\mathbf{x}_i, \mathbf{f}_{i-1})$ regardless of whatever the $\mathbf{x}_i$ is. However, given different $\mathbf{x}_i$, the attention model is able to produce different context weights given different input query vector $\mathbf{q}_i$ (since $\mathbf{q}_i = \mathbf{x}_i$ in Eq. 3.5); i.e., the contexts will be weighted accordingly for different $\mathbf{x}_i$, so that any context can only be effectively used when the attention model detects its relevance and gives it a high weight by Eq. 3.4 and 3.5. The alignment model (Eq. 3.5) is explicitly learned as a selector to dynamically detect relevant and useful contexts over irrelevant ones.

However, as yet, given a fixed input query $\mathbf{q}_i$, the alignment model of Eq. 3.5 equally considers the other input features $\mathbf{x}_j$, for each $j = 1 \dots n$, in the attention computation, remaining unaware of that the segment i is being the target one to classify. Therefore, inspired by the position-based gating procedure in [81], the scores e_{ij} can be penalized based on the relative position of the context segment j and target i before being normalized to weight α_{ij}:

$$\alpha_{ij} = \frac{d(i,j)\exp(e_{ij})}{\sum_{k=1}^{n} d(i,k)\exp(e_{ik})}, \quad \forall j = 1 \dots n \tag{3.6}$$

where $d(i,j)$ is a gating function of one hidden layer NN and logistic sigmoid output ($[0,1]$):

$$d(i,j) = \begin{cases} 1, & j = i \\ \sigma_2(w_2\sigma_1(w_1|i-j| + b_1) + b_2), & \forall j \neq i \end{cases} \tag{3.7}$$

where σ_1 is an activation function (tanh), σ_2 a sigmoid function, and w_1, w_2, $b_1, b_2 \in \mathbb{R}^1$. Such additional gating procedure helps favor the weight of target $\mathbf{x}_i$ and penalize the effects of any contexts far from the target, so that it can

presumably prevent $\mathbf{c}_i$ (Eq. 3.3) from being overwhelmed by context segments regardless of the target $\mathbf{x}_i$.

3.5 Experiments

3.5.1 Experimental setup

3.5.1.1 Data

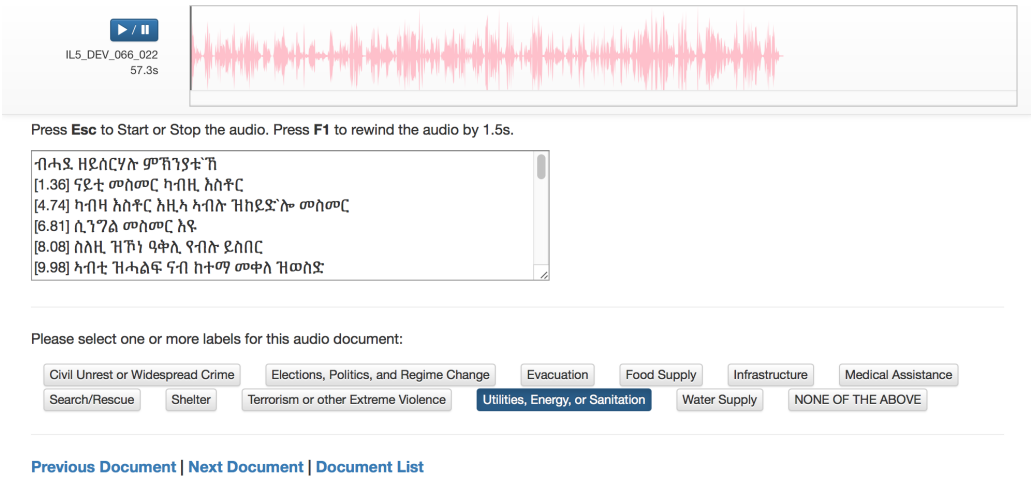
The LORELEI Situation Frame (SF) task is characterized by extremely limited training resources. The only available resources for each evaluation language, called an Incident Language (IL) are:

- Monolingual text, some of which might be related to the incident.
- Untranscribed, unlabeled audio.
- A small amount of IL-English parallel text.
- Optionally, a few hours of consultation with a native informant (NI).

The NI is a native speaker of the IL with at least intermediate proficiency in English. System developers may ask the NI to perform any annotation tasks deemed necessary to build a system for identifying SFs from speech, e.g. transcribing some IL speech or labeling segments with SF topic labels. To increase the NI's annotation efficiency, all NI tasks were conducted via a web browser-based user interface tailored to the specific LORELEI tasks, as shown in Figure 3.2.

The dev and eval datasets we used are as shown in Table 3.3. For Turkish, Arabic, Spanish and English, each language is a single dataset and seen as

Figure 3.2: NI user interface optimized for speech transcription and SF Type labeling.



dev set. Their topic label annotations for all segments are given, and used for training the topic ID classifiers[3].

For Mandarin, Tigrinya and Oromo, each language has one DEV and EVAL set respectively; true topic labels on these DEV sets are unavailable, so we selected some segments, collected their hypothesized topic labels from NI, and included them into the classifier training. Also on these DEV sets, we selected some segments for the NI to transcribe and used them for ASR adaptation. The total given NI session for consultation was 2 hours for Mandarin, 10 hours each for Tigrinya and Oromo. Only on Tigrinya and Oromo DEV sets, we collected transcribed speech from the NI, 27 mins and 18 mins respectively. The EVAL sets of these three languages, in addition to the single Russian dataset, are provided with true topic annotations and are used for evaluating the system performance.

[3]Since Spanish set is overwhelmed by the segments of topic “Elections and Politics", we filtered out all segments that include that topic.

Dataset	Language Pack	LDC Catalog	$\lvert\mathbb{D}_{doc}\rvert$	$\lvert\mathbb{D}_{seg}\rvert$	Topic Label	ASR Corpora
Dev	Turkish	LDC2016E109	212	2095	LDC	BABEL [34]
	Arabic	LDC2016E123	47	1025	LDC	GALE [82]
	Spanish	LDC2016E127	198	393	LDC	HUB4 [83]
	US English	LDC2017E50	154	842	LDC	–
	Mandarin DEV	LDC2016E108	77	100	NI	GALE [84]
	Tigrinya DEV	LDC2017E35	130	159	NI	Universal
	Oromo DEV	LDC2017E36	241	364	NI	Universal
Eval	Mandarin EVAL	LDC2016E115	119	724	LDC	GALE [84]
	Russian	LDC2016E111	136	787	LDC	Universal
	Tigrinya EVAL	LDC2017E37	116	1095	LDC	Universal
	Oromo EVAL	LDC2017E38	46	709	LDC	Universal

Table 3.3: LORELEI speech data description. $\lvert\mathbb{D}_{doc}\rvert$ denotes the number of documents. $\lvert\mathbb{D}_{seg}\rvert$ denotes the number of segments. Manual transcripts are provided for US English corpus. 'Universal' refers to the universal phone set ASR described in Section 3.3.

In sum, when evaluating on Mandarin EVAL or the Russian dataset, the training data for learning topic ID models consists of Turkish, Arabic, Spanish, US English and Mandarin DEV. When evaluating on Tigrinya EVAL or Oromo EVAL, we use the same training data in addition to Tigrinya DEV or Oromo DEV, respectively.

3.5.1.2 Evaluation metrics

Under the LORELEI Speech SF evaluation framework as described in [51], topic ID system outputs are evaluated in two layers using average precision (AP, equal to the area under the precision-recall curve).

The *Relevance* layer is to separate the segments with at least 1 in-domain topic from non-relevant out-of-domain segments. Specifically, each segment is given 11 posteriors over each in-domain topic, and the Relevance scorer takes

the maximum one as the in-domain posterior. Thus, given each confidence threshold, the scorer can compute the precision and recall by comparing against the true binary in-/out-of-domain label. Finally, given the resulting precision-recall points for each threshold, the scorer computes the area under the precision-recall curve, i.e. AP, as the Relevance layer score.

The *Type* layer is to detect all present 11 in-domain topics. First, for a given confidence threshold, Type scorer computes the micro-averaged precision and recall across 11 in-domain topics, which calculates precision and recall globally by counting the total true positives, false positives and false negatives across 11 in-domain topics (i.e., giving equal importance to each data instance). Then, given the micro-averaged precision and recall at each evaluated threshold, the scorer computes the AP, as the Type layer score.

3.5.1.3 ASR

Audio transcripts exist only for the LORELEI English speech dataset. For the Turkish, Arabic, Spanish and Mandarin datasets, we used preexisting transcribed speech corpora, as shown in Table 3.3, to train ASR systems with Kaldi [47], and then decoded the LORELEI datasets using the appropriate ASR. For Russian, Tigrinya and Oromo, transcribed speech corpora were unavailable and we used the universal phone set ASR to decode each corpus, by rebuilding the decoding graph using a new pronunciation lexicon and language model trained on the monolingual texts in the LORELEI datasets.

For experiments on Tigrinya and Oromo, we use a selection of 10 BABEL languages for ASR training ($\sim$600h), 7 of which were chosen as in [12], with

3 more chosen arbitrarily (Guarani, Mongolian, Dholuo). We bootstrapped the lexicon using a G2P trained on a seed lexicon derived from the provided resources. For Tigriyna the seed was a dictionary of words with IPA pronunciations, and for Oromo the seed was an approximate grapheme-to-phoneme map. The vocabulary (word list) was generated from the provided monolingual text. We (re)normalized the text according to IL specific punctuation rules. Additional sources of words were the bilingual gazetteer and transcripts obtained during the NI sessions. The language model was trained on the same text. Language model hyper-parameters were chosen to minimize perplexity on a held-out set (i.e. small subset of the monolingual text not used for training).

For Russian, we use 10h subsets of 21 BABEL languages ($\sim$200h) in training (all except Haitian, Vietnamese, Amharic, Georgian). This reduces training time, provides better phoneme coverage, and performs as well or better in word error rate as the 10-language ASR above on the BABEL Haitian, Amharic and Georgian dev sets. For Russian, we used `wikt2pron`[4] to generate a seed lexicon by scraping Wiktionary for XSAMPA pronunciations of all Russian words found in the provided monolingual text. We also filtered out all words not written in Cyrillic, and to discard apparent misspellings, we used only the 600k most frequent remaining words.

Note that speech segment lengths vary between 5 seconds and 2 minutes, with an average duration of about one minute. Since ASR systems have difficulty decoding long segments, we further segmented the audio using either

[4]https://github.com/abuccts/wikt2pron

the overlapped segmentation approach as in [85], or voice-activity-detection (by a DNN-based speech activity detection system that segmented audio into speech and silence). For the overlapped segmentation, we used chunks 15 seconds long repeated every 10 seconds and then filtered the transcripts by removing words whose midpoints were within 2.5 seconds to the chunk edge before combining them into a single transcript.

In addition, we trained two Gaussian mixture models (GMMs) on the speech and music portions of MUSAN [77]. Each speech segment is split into 15 second chunks but without overlap. Then for each chunk, two average frame-level log-likelihoods were calculated by the music and speech GMMs respectively, to further produce a music-to-speech log-likelihood ratio γ. γ went through a sigmoid function and produced a posterior score. Finally for each speech segment, we used the maximum posterior score over all chunks as the music posterior feature δ for that segment, which was then concatenated to the LSA features (Section 3.4.1).

3.5.1.4 MT

Supervised topic label information in various languages can all be projected into English topic classifiers through bilingual (i.e., foreign language to English) translation lexicons. Each bilingual MT table was derived from the parallel training data with words aligned automatically by the GIZA++ [86] and Berkeley aligner [87], independently under the MT effort. Any preexisting training data can be used in addition to the data provided by the LORELEI program.

We translated each foreign word in the ASR transcript into its four most likely English translations. Then we mapped any unicode data into their nearest ASCII characters, and filtered stop words using the lists from [79, 88], and any words with three or fewer characters.

3.5.1.5 Classification models

First, the tf-idf or LSA features were learned as described in Section 3.4.1. For the four eval languages overall, we found LSA dimensions over $\{300, 600, 900\}$ can generally produce improvements over tf-idf features, and the ones we finally used are shown in Table 3.4.

The non-contextual SVM and NN were learned as in Section 3.4.2. Contextual RNN and attention based models are described in Section 3.4.3 and 3.4.4 respectively. Also, validation data is needed for model parameter tuning and during NN training. While evaluating Mandarin, we left a small portion out of the training data as validation data. While evaluating Tigrinya, Oromo and Russian, we used the Mandarin EVAL dataset as validation data. The performance of SVMs did not vary much after 30 SGD epochs. While each NN-based model was trained for up to 50 epochs, the model with the best accuracy on the validation data was used for evaluation on the eval data. For each experiment, we repeated it 5 times, and the means are reported in Table 3.5 (standard deviation is omitted for clarity).

Some parameters were tuned and shared for all languages. SVMs used ℓ_2 regularization constant 0.001 on tf-idf features. All NN-based models had hidden layer size 512 and rectified linear unit (ReLU) nonlinearities, and were

Table 3.4: Differing topic ID model parameters across eval languages.

Eval language	Russian	Mandarin	Tigrinya	Oromo
LSA dimension	300		900	
SVM ℓ_2 regularization constant	0.001	0.0001		
# hidden layers in NN	1	2		
# hidden layers in RNN	0	1		
# hidden layers in attention-based	1	2		
Dropout rate	0.5	0.25		

trained with Adam optimizer [89]. Non-contextual NN used mini-batch size of 256 spoken segments. Contextual RNN or attention based models used the mini-batch size of 6 spoken documents. For RNN-based models, we found GRU slightly outperformed the conventional Elman RNN or LSTM, and we used the GRU layer that took the LSA features as inputs. All neural network-based models were implemented with PyTorch [90].

The remaining parameters were the same when evaluating Mandarin, Tigrinya and Oromo, but differed for Russian, as shown in Table 3.4. When evaluating Russian, we found using SVM ℓ_2 regularization constant 0.001 on LSA features, one NN hidden layer and dropout rate 0.5 gave much better results instead; presumably because the universal phone set ASR for Russian was unadapted, the resulting transcripts were more noisy and required stronger regularization. Also, we used one GRU layer directly followed by the output layer. Each contextual vector $\mathbf{c}_i$ (Section 3.4.4) was followed by one hidden layer instead of two. Note that we used the above model parameters different from other three eval languages to obtain optimal results for both Russian non-contextual and contextual models, so that the comparisons between the two are fair. In other words, within each eval language, we focus on

Table 3.5: Topic classification results on LORELEI speech datasets, evaluated by the average precisions of Type layer and Relevance (Rel) layer (Section 3.5.1.2). LSA_δ is each LSA feature vector concatenated with music posterior δ. Attn^1 or Attn^2 is each attention-based contextual model that uses 1 or 2 nearest context segments, respectively. Attn^1_{pos} or Attn^2_{pos} denotes that the additional position-based gating procedure in attention model is enabled. Last row shows the 10-fold cross-validation results on each eval set using ASR transcripts and true topic labels (without using MT or any other dev set), as oracle results for comparison.

	Mandarin		Russian		Tigrinya		Oromo		Average	
Model	Type	Rel	Type	Rel	Type	Rel	Type	Rel	Type	Rel
tf-idf, SVM	0.458	0.702	0.382	0.854	0.371	0.554	0.382	0.772	0.398	0.721
LSA, SVM	0.505	0.739	0.386	0.856	0.392	0.561	0.409	0.782	0.423	0.735
LSA_δ, SVM	0.510	0.742	0.408	0.870	0.422	0.600	0.423	0.822	0.441	0.759
LSA_δ, NN	**0.519**	**0.743**	**0.415**	**0.881**	**0.451**	**0.625**	**0.436**	**0.819**	**0.455**	**0.767**
LSA_δ, RNN	0.525	0.737	0.430	0.894	0.389	0.578	0.467	0.820	0.453	0.757
LSA_δ, Attn^1	**0.544**	**0.741**	**0.466**	**0.888**	0.407	0.597	**0.495**	**0.828**	0.478	0.764
LSA_δ, Attn^1_{pos}	0.542	0.744	0.449	0.884	**0.455**	**0.618**	0.482	0.830	**0.482**	**0.769**
LSA_δ, Attn^2	0.537	0.742	0.461	0.892	0.365	0.557	0.494	0.838	0.464	0.757
LSA_δ, Attn^2_{pos}	0.543	0.746	0.448	0.887	0.444	0.611	0.491	0.831	**0.482**	**0.769**
10-fold CV	*0.576*	*0.843*	*0.444*	*0.838*	*0.574*	*0.719*	*0.419*	*0.750*	*0.503*	*0.788*

drawing fair comparisons between its optimal non-contextual and contextual models.

3.5.2 Non-contextual topic classification results

Table 3.5 first shows the results based on non-contextual model SVM and NN. The LSA transformation on tf-idf features substantially improved performance across the board, and also mapped the high-dimensional tf-idf vectors (around 25k) to a dimension small enough for the LSA features to be used as inputs to NN-based models. Additionally, appending auxiliary music posteriors (Section 3.4.1) to the LSA features can produce large gains, except on Mandarin; we found for the Mandarin dataset music was less indicative of out-of-domain

topics. Finally, feedforward NNs were generally more competitive than linear SVMs when using the same input LSA features.

3.5.3 Contextual topic classification results

Table 3.5 further shows the results of our experiments using the proposed contextual RNN and attention models. The GRU-based contextual models outperformed the best non-contextual NN models on Russian and Oromo, but not on Mandarin or Tigrinya. For Mandarin, we had a high-performing ASR system trained on around 600 hrs of transcribed speech from GALE [84], so the Mandarin transcripts were much more accurate than other languages, which presumably made it more difficult to improve the non-contextual baseline results; inference from contexts might be helpful to recover the ASR errors in the target segment, and thus better ASR transcripts often allow for confident classification without having to consider additional contexts. For Tigrinya EVAL set, we found around 72% of the segments were out-of-domain; i.e., if a target segment is mostly surrounded by out-of-domain segments, using its contexts can give adverse effects, and the overall results can be worse than the context-independent counterparts.

We further experimented with contextual attention based models, using the contexts of 1 or 2 nearest left and right segments, i.e. when $L = R = 1$ or $L = R = 2$ in Section 3.4.4. The attention-based models outperformed the non-contextual models, except on Tigrinya, due to the overwhelming amount of out-of-domain segments, as discussed above. However, we can match the performance of the non-contextual models on Tigrinya, with only a

small performance loss in the other languages, by using the additional gating mechanism in Eq. 3.6. The gating mechanism partially penalizes the context effects and makes the model aware of the target segment location. Note that, the attention-based models consistently outperformed the RNN-based models, and it demonstrates the efficacy of the gated attention mechanism that dynamically selects and uses more relevant contexts instead of receiving contexts in a deterministic manner.

Overall, with respect to the best context-independent models, the contextual attention based models produced comparable performance on Tigrinya, and produced considerable performance improvements on the rest three eval languages. Also, the results of using wider contexts, i.e., 2-nearest left and right segments, were comparable to those of using 1-nearest only. In addition, the attention function we used in Eq. 3.5 is also called additive attention, and we found it outperformed the dot-product (multiplicative) attention [91]. We also experimented with multi-head attention [91] and component (or multi-dimensional) attention [92], but none of these techniques can give us better results, presumably due to the small size of our topic model training data.

3.5.4 Ten-fold cross validation analysis

So far, we have only used English translations of each dev and eval language to resolve the language mismatch, but the training and eval datasets can be severely mismatched. An oracle result against which we can compare is the 10-fold cross validation (CV) performance on each eval set itself, where each experiment uses part of the true eval set topic labels for training. For each

eval language, we split the corresponding eval set into 10 folds, used the extracted LSA features over the raw ASR transcripts (without translation or any data from other language), completed 10 monolingual supervised SVM classifications with true topic labels, and reported the average of each 10 experiments as shown in the last row of Table 3.5.

For each language, such 10-fold CV results give us estimates of the topline numbers we could achieve with around 700 in-domain training exemplars. First, the gap between each topline number and the full accuracy (i.e. AP = 1) mostly indicates the given ASR quality and the intrinsic difficulty of each eval dataset. Next, comparing our cross-lingual approach with such monolingual topline, we found using the above contextual topic ID approach had reduced the gap on Mandarin, and surpassed the topline on Russian and Oromo, while falling behind on Tigrinya (due to the train-test discrepancy in the amount of out-of-domain segment occurrences as discussed in Section 3.5.3).

3.6 Conclusion

In classifying spoken documents into predefined classes, audio documents collected in the wild can be extremely long and contain multiple class label shifts (e.g. topic shifts) at varying locations in the audio, so we need to perform classification on a sequence of segmented audio. Each resulting speech segment is of reasonable length and semantically self-contained, such that each of them can be independently classified. We first presents a general classification system that combines universal acoustic modeling, evaluation language to English machine translation and an English-language classifier.

This combination requires no transcribed speech in the evaluation language, leading to near language-agnostic operation.

Furthermore, we have performed comprehensive experiments on the LORELEI datasets in a realistic low-resource scenario, and have found that, exploiting the context segments can provide considerable topic classification performance improvements over the context-independent models. Finally, comparing our contextual modeling frameworks, we demonstrate that the proposed attention-based models which leverage context segments in a selective approach can consistently outperform the RNN-based alternatives.

Chapter 4

Spoken Document Classification without ASR

In the preceding Chapter 3 we have introduced the modern spoken document classification systems that typically use automatic speech recognition (ASR) to produce speech transcripts, and perform classification on ASR outputs by supervised training of classifiers. While under resource-limited conditions with little or no transcribed speech annotations for a language of interest, Chapter 3 has demonstrated an universal phone set ASR to produce adequate speech transcripts that can effectively enable the subsequent classification task. However, it still requires monolingual text and pronunciation lexicons from that language to start the processing. In this chapter[1], we further explore an alternative line of approach to decoding speech that removes the above needs, using unsupervised speech technologies of lexical discovery and phonetic discovery.

[1]Large portions of this chapter have been published in [93, 67]

4.1 Introduction

To date it is very challenging to model many world's low-resource languages, most of which are under-documented or unwritten. According to UNESCO (United Nations Educational, Scientific and Cultural Organization), 80 percent of African languages have no orthography[2] [95], and thus no written record. In practice, many speakers of such endangered languages are bilingual or multilingual, so collecting other linguistic annotations, such as spoken translations or document-level topics, can be more feasible in the absence of an orthographic lexicon.

After sourcing the recorded speech, we also need to transform the raw speech data into a format that can be efficiently indexed and searched. Typically, this format is based on orthographic word, and the transformation process is automatic speech recognition. However, we can consider ASR as one of the many ways to transform acoustic signals into written tokens, and we refer to the general transformation process as *tokenization*, so that the fixed set of tokens used to characterize speech can be of any type, such as orthographic word or any smaller unit like phoneme. As a result, speech is *tokenized* into sequences of tokens, on which the subsequent indexing and retrieval is performed.

Developing general tokenization approaches is particularly useful in the realistic scenario, where the orthographic lexicon of a language is unavailable or

[2] An orthography is a set of conventions for writing a language, which includes norms of spelling, hyphenation, capitalization, word breaks, emphasis, and punctuation [94].

nonexistent so that the supervised training of a standard ASR system is infeasible. In such case, previous work demonstrates that the language-mismatched phoneme recognizers can produce cross-lingual tokenizations effectively for topic classification [96, 97], but the performance is highly dependent on the level of language mismatch and environmental condition mismatch (channel, noise, etc.) between the training and testing datasets.

Alternatively, in this chapter, we focus on unsupervised tokenization approaches that operate directly on the speech of interest.

4.2 Related work

First, unsupervised term discovery (UTD), sometimes also referred to as 'lexical discovery' or 'spoken term discovery', is one such approach that aims to identify and cluster repeating word- or phrase-like patterns across speech [98]. Each resulting cluster represents a discovered word type (i.e. a distinct lexical entry), and speech can be characterized with these hypothesized word categories. Most UTD systems are based around segmental dynamic time warping (DTW) [99, 100, 101], and recent work [98] presents a novel unsupervised Bayesian framework that jointly segments speech into word-like segments and clusters these segments into hypothesized words.

Thus UTD provides a way of automatically detecting indexable terms via acoustic repetition, and the indexing terms identified by UTD have been shown to be effective in spoken document classification [54], spoken document retrieval [102], and interactive exploration of speech collections [103]. However, the classification results in [54] are limited since the acoustic features

on which UTD is performed are produced by acoustic models trained from the transcribed speech of its evaluation corpus. In this chapter, we further investigate UTD-based document classification performance when UTD operates on language-independent speech representations extracted from multilingual bottleneck networks trained on languages other than the evaluation language.

Another unsupervised tokenization alternative is phonetic discovery, also known as acoustic unit discovery (AUD), which is the process of automatically identifying the categorical subword or phonemic inventory and relating it to the underlying acoustics [104]. Thus far most existing methods focus on unsupervised learning of hidden Markov model (HMM) based phoneme-like units from untranscribed speech, where each HMM represents an induced acoustic unit. For example, [96] presents an approach to initialize the unsupervised HMM training with the label sequences produced by segmental Gaussian mixture models (GMMs), using maximum likelihood parameter estimations. [105] formulates a Dirichlet process mixture model where each mixture is a GMM-HMM based acoustic unit, using Bayesian inference via Gibbs sampling. To scale computationally to large speech datasets, [106] applies the Variational Bayesian inference to the Dirichlet process mixture model, which allows for parallelized large-scale training. [93] further extends [106] to a context-rich framework by the self-supervised linear discriminant analysis that incorporates phonetic contexts into the front-end acoustic features.

Recent success in deep generative modeling, such as deep belief network, generative adversarial network, etc., motivates a new AUD framework that

composes the latent graphical models, i.e. HMMs, with neural network observation likelihoods, known as variational autoencoder HMM (VAE-HMM), or structured VAE [107, 108, 109]. While HMMs are still used as the structured dynamics models, acoustic observations (e.g. MFCCs) are first mapped to a latent space and the resulting latent representations are then modeled by the Gaussians specific to each HMM state. [107] estimates the parameters of each state-specific Gaussian (i.e. the priors of the latent space) with maximum likelihood estimation, while [108, 109] use Gaussians with conjugate prior. However, [107, 108, 109] all limit VAE to reconstructing each acoustic frame individually, and the latent representation of each frame is independent of the context frames. In this chapter we aims to develop a context-dependent VAE that infers a context-rich latent representation from each set of stacked frames.

Thus far we have three different methods to tokenize speech using indexing tokens – orthographic words decoded by ASR, word-like units detected by UTD, and phoneme-like units identified by AUD. Further, in performing spoken document classification on these various tokenizations, prior works [54, 110, 55, 96, 97, 93] are limited to using bag-of-words features as document representations. While UTD mostly aims to identify relatively long (0.5 - 1 sec) repeated terms, ASR/AUD enables full-coverage segmentation of each continuous speech utterance into a sequence of words/units, and such resulting temporal sequence enables another feature learning architecture based on convolutional neural network (CNN) [59]; instead of treating the sequential tokens as a bag of words or acoustic units, the whole token sequence is encoded as concatenated continuous vectors, and followed by

convolution and temporal pooling operations that capture the local and global dependencies. Such continuous space feature extraction frameworks have been explored in various language processing tasks such as part-of-speech tagging [59, 60], spoken language understanding [111, 69], and text document classification [61, 112]. However, three questions are worth investigating in our AUD-based tokenizations:

i. If such a CNN-based framework can perform as well on noisy automatically discovered phoneme-like units as on orthographic words/characters.

ii. If pre-trained vectors of phoneme-like units from *word2vec* [113] provide superior performance to random initialization as evidenced by the word-based tasks.

iii. If CNNs are still competitive in low-resource settings of hundreds to two-thousand training exemplars, rather than the large/medium sized datasets as in previous work [61, 112].

Thus, incorporating different tokenization, i.e. UTD, AUD and ASR, and different document feature representation approaches noted above, we perform comprehensive evaluations on both single-label and multi-label spoken document classification tasks, and investigate how the performances compare accordingly.

4.3 Unsupervised term discovery

UTD aims to automatically identify and cluster the repeated terms (e.g. words or phrases) from speech, via acoustic repetitions. To circumvent the exhaustive

DTW-based search limited by $\mathcal{O}(n^2)$ time [99], [100] proposed a scalable UTD framework which permits search in $\mathcal{O}(n \log n)$ time, and implemented it in the Zero Resource Toolkit (ZRTools). In this section, we briefly outline the essentials of [100] and describe the UTD procedures in ZRTools by four steps below:

1. Construct the sparse approximate acoustic similarity matrices between pairs of speech utterances.

2. Identify word repetitions via fast diagonal line search and segmental DTW.

3. The resulting matches are used to construct an acoustic similarity graph, where nodes represent the matching acoustic segments and edges reflect DTW distances.

4. Threshold the graph edges, and each connected component of the graph is a cluster of acoustic segments, which produces a corresponding term (word/phrase) category.

Finally, the cluster of each discovered term category consists of a list of term occurrences.

Note that in the third step above, the weight on each graph edge can be exact DTW-based similarity, or other similarity based on heuristics more than DTW distance. For example, we investigate an implementation in ZRTools, where a separate logistic regression model is used to rescore the similarity between identified matches by determining how likely the matching pair is the same underlying word/phrase and is not a filled pause (e.g. "um-hum" and

"yeah uh-huh" in English). Filled pauses tend to be acoustically stationary with more phone repeats and thus would match throughout the acoustic similarity matrix, whereas a contentful word (without too many phone repeats) tend to concentrate around the main diagonal; thus, the features in logistic regression contain the numbers of matrix elements in diagonal bands in progressive steps away from the main diagonal. Feature weights are learned using a portion of transcribed speech with reference transcripts, and the resulting model can be used for language-independent rescoring.

4.4 Acoustic unit discovery

In this section, we first briefly describe the variational Bayesian inference based AUD framework in [106], and then describe the VAE-HMM based AUD in [107]. Next, we present our extended variant, referred to as a structured contextual VAE or contextual VAE-HMM. Finally we discuss experimental results on the intrinsic measurements of our AUD performance.

4.4.1 GMM-HMM

As presented in [106], a phone-loop model is formulated where each phoneme-like unit is modeled as an HMM with GMM output density (GMM-HMM), as illustrated in Figure 4.1. Under the Dirichlet process framework, we consider the phone loop as an infinite mixture of GMM-HMMs, and the mixture weights $\{\pi_m\}_{m=1}^{\infty}$ are based on the stick-breaking construction of Dirichlet process. The infinite number of units in the mixture is truncated by some large count M in practice, giving zero mixture weight to any unit beyond M, i.e.

$\{\pi_m\}_{m=1}^{M}$.

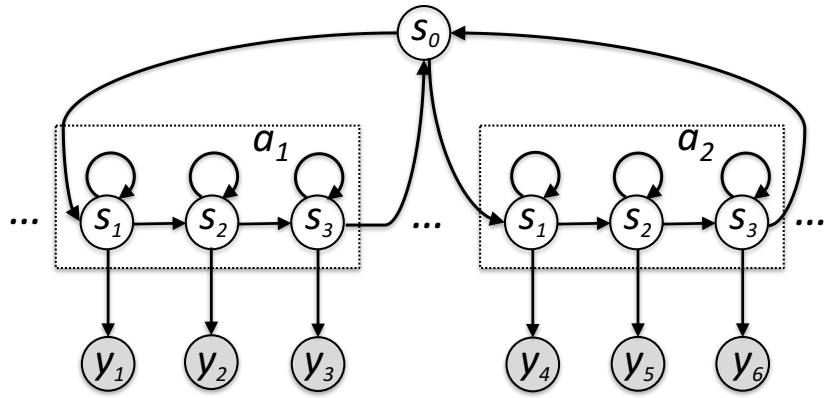


Figure 4.1: An illustration of the directed graphical model as an infinite phone-loop AUD model. a_1 and a_2 denote the acoustic unit 1 and 2. s_i, for each $i = 1 \ldots 3$, denotes an HMM state.

Following the variational Bayesian inference, we aim to infer both the latent variables **H** (i.e., the indices of HMM, HMM state, and GMM component), and the unknown generative model parameters θ (i.e., GMM/HMM parameters). The detailed update equations can be found in [106]. We treat such mixture of GMM-HMMs as a single unified HMM and thus the segmentation of the data is performed using standard forward-backward algorithm. Training is fully unsupervised and parallelized across utterances. After a fixed number of training epochs, we use Viterbi decoding algorithm to obtain acoustic unit tokenizations of the data, i.e., $\mathbf{a} = a_1, ..., a_n$.

4.4.2 Structured VAE

We first briefly describe the variational inference and then present detailed theoretical derivations of the VAE-HMM framework in [107].

4.4.2.1 Variational inference

Consider the observations $\mathbf{Y} = \{\mathbf{y}_t\}_{t=1}^{T}$ consisting of T samples of a continuous variable $\mathbf{y}$. We assume $\mathbf{Y}$ is generated by some random process involving the hidden variables $\mathbf{H}$. Variational inference uses the distribution $q(\mathbf{H}|\mathbf{Y};\phi)$, parameterized by the variational parameters ϕ, to approximate the intractable true posterior $p(\mathbf{H}|\mathbf{Y};\theta)$, where θ is known as generative model parameters. The marginal log-likelihood can be written as:

$$\log p(\mathbf{Y};\theta) = D_{\mathrm{KL}}(q(\mathbf{H}|\mathbf{Y};\phi)\|p(\mathbf{H}|\mathbf{Y};\theta)) + \mathcal{L}(\mathbf{Y};\theta,\phi) \tag{4.1}$$

where D_{KL} denotes the Kullback–Leibler (KL) divergence, and

$$\mathcal{L}(\mathbf{Y};\theta,\phi) = \mathbb{E}_{q(\mathbf{H}|\mathbf{Y};\phi)}\left[\log p(\mathbf{Y}|\mathbf{H};\theta)\right] - D_{\mathrm{KL}}(q(\mathbf{H}|\mathbf{Y};\phi)\|p(\mathbf{H};\theta)) \tag{4.2}$$

Since D_{KL} is always non-negative and $\log p(\mathbf{Y};\theta) \geq \mathcal{L}(\mathbf{Y};\theta,\phi)$, $\mathcal{L}(\mathbf{Y};\theta,\phi)$ is called the variational lower bound on the marginal likelihood of the data $\mathbf{Y}$. We aim to optimize the lower bound of Eq. 4.2 and it can be done by the Expectation–Maximization (EM) algorithm by alternating between:

i. E-step: infer $q(\mathbf{H}|\mathbf{Y};\phi)$ to approximate $p(\mathbf{H}|\mathbf{Y};\theta)$.

ii. M-step: maximize the lower bound $\mathcal{L}(\mathbf{Y};\theta,\phi)$ with respect to both the variational parameters ϕ and generative parameters θ.

4.4.2.2 VAE

We first briefly describe VAE. As above, we consider one speech utterance characterized by $\mathbf{Y} = \{\mathbf{y}_t\}_{t=1}^{T}$ as a sequence of observations. A D_y dimensional

vector $\mathbf{y}_t$ is modeled by a D_x dimensional latent vector $\mathbf{x}$ through a non-linear transformation $f(\mathbf{x};\gamma)$ with parameters γ:

$$p(\mathbf{y}_t|\mathbf{x};\gamma) = \mathcal{N}(\mathbf{y}_t; f(\mathbf{x};\gamma), \sigma_y^2 \mathbf{I}_{D_y}) \tag{4.3}$$

where $f(\mathbf{x};\gamma)$ is given by a neural network which is referred to as a probabilistic decoder, σ_y a constant[3], and $\mathbf{I}_{D_y}$ is a D_y dimensional identity matrix. The latent variable $\mathbf{x}$ is assumed to be generated by a normal distribution:

$$p(\mathbf{x};\theta) = \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) \tag{4.4}$$

The unobserved variable $\mathbf{x}$ is also called latent representation or code [114]. To approximate the true $p(\mathbf{x}|\mathbf{y}_t;\theta)$, we let the variational approximate posterior $q(\mathbf{x}|\mathbf{y}_t;\phi)$ be a multivariate Gaussian with mean vector $\boldsymbol{\mu}_t = [\mu_{t,1},\ldots,\mu_{t,D_x}]^T$ and diagonal covariance matrix $\boldsymbol{\Sigma}_t = \text{diag}(\sigma_{t,1}^2,\ldots,\sigma_{t,D_x}^2)$ that are given by the transformation $g(\mathbf{y}_t;\phi)$:

$$(\boldsymbol{\mu}_t; \log \boldsymbol{\Sigma}_t) = g(\mathbf{y}_t;\phi) \tag{4.5}$$

such that

$$q(\mathbf{x}|\mathbf{y}_t;\phi) = \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t) \tag{4.6}$$

where $g(\mathbf{y}_t;\phi)$ is a neural network with variational parameters ϕ, referred to as probabilistic encoder. Thus, $\mathbf{Y}$ and $\mathbf{x}$ along with θ and ϕ form a VAE.

[3]We use a constant σ_y instead of modeling it with another decoder.

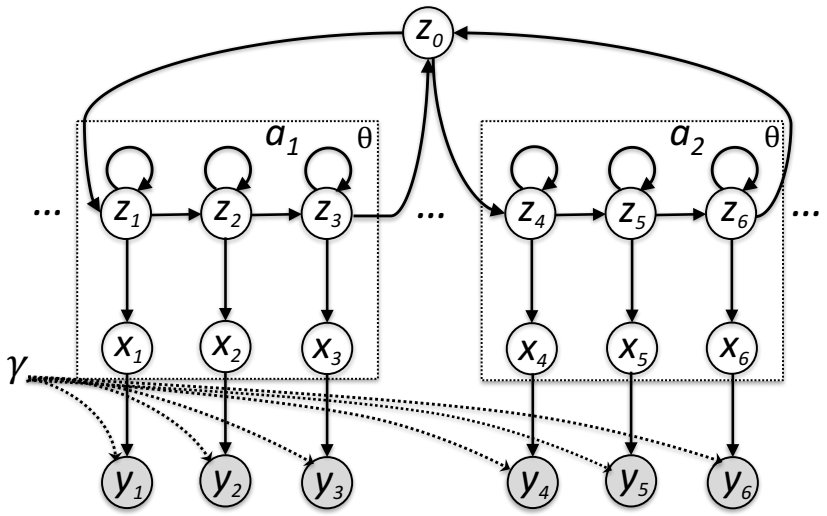


Figure 4.2: An illustration of the directed graphical model as VAE-HMM. z_i denotes the latent HMM state, x_i the latent representation, y_i the observation.

4.4.2.3 VAE-HMM

Note that in standard VAE, each latent representation $\mathbf{x}_t$ is independent of each other, being drawn from Eq. 4.4. To model the temporal dynamics, we compose the VAE with HMMs, referred to as VAE-HMM or structured VAE, as illustrated in Figure 4.2. Each of the U distinct discovered acoustic units ($U \leq M$ with M as the truncation level in Section 4.4.1) is modeled by a 3-state HMM with standard left-to-right typology. Thus, each latent representation $\mathbf{x}_t$ is generated by a latent state variable z_t, through a state specific normal distribution:

$$p(\mathbf{x}_t|z_t = k; \theta) = \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4.7}$$

where $K = 3U$, and $\theta = \{\{\pi_u\}_{u=1}^{U}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}\}$ is the set of generative model parameters; $\mathbf{Z} = \{z_t\}_{t=1}^{T}$ are related through a Markov process, which control the HMM state (i.e., acoustic unit) to be selected for each representation

$\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^T$.

VAE-HMM: E-step inference. Given the conditional independence assumptions in directed graphical models, we have

$$p(\mathbf{X}, \mathbf{Z}, \mathbf{Y}; \theta, \gamma) = p(\mathbf{Y}|\mathbf{X}, \mathbf{Z}; \theta, \gamma)p(\mathbf{X}, \mathbf{Z}; \theta) = p(\mathbf{Y}|\mathbf{X}; \gamma)p(\mathbf{X}|\mathbf{Z}; \theta)p(\mathbf{Z}; \theta) \quad (4.8)$$

The following mean field approximation gives:

$$\begin{aligned}
&\log q(\mathbf{Z}|\mathbf{Y}; \theta, \gamma, \phi) \\
&= \mathbb{E}_{q(\mathbf{X}|\mathbf{Y};\phi)} [\log p(\mathbf{X}, \mathbf{Z}, \mathbf{Y}; \theta, \gamma)] + const \\
&= \mathbb{E}_{q(\mathbf{X}|\mathbf{Y};\phi)} [\log p(\mathbf{Y}|\mathbf{X}; \gamma) + \log p(\mathbf{X}|\mathbf{Z}; \theta) + \log p(\mathbf{Z}; \theta)] + const \\
&= \mathbb{E}_{q(\mathbf{X}|\mathbf{Y};\phi)} [\log p(\mathbf{X}|\mathbf{Z}; \theta)] + \log p(\mathbf{Z}; \theta) + const \\
&= \sum_{t=1}^{T} (\mathbb{E}_{q(\mathbf{x}_t|\mathbf{y}_t;\phi)} [\log p(\mathbf{x}_t|z_t; \theta)] + \log p(z_t|z_{t-1}; \theta)) + const
\end{aligned} \quad (4.9)$$

where *const* is a normalizing constant. Then by the definition of KL divergence,

$$\begin{aligned}
&\mathbb{E}_{q(\mathbf{x}_t|\mathbf{y}_t;\phi)} [\log p(\mathbf{x}_t|z_t; \theta)] \\
&= \mathbb{E}_{q(\mathbf{x}_t|\mathbf{y}_t;\phi)} [\log q(\mathbf{x}_t|\mathbf{y}_t; \phi)] - D_{\mathrm{KL}}(q(\mathbf{x}_t|\mathbf{y}_t; \phi) \| p(\mathbf{x}_t|z_t; \theta))
\end{aligned} \quad (4.10)$$

where

$$\begin{aligned}
\mathbb{E}_{q(\mathbf{x}_t|\mathbf{y}_t;\phi)} [\log q(\mathbf{x}_t|\mathbf{y}_t; \phi)] &= -H(\mathbf{x}_t|\mathbf{y}_t) \\
&= -\frac{1}{2}(D_x + D_x \log(2\pi) + \sum_{j=1}^{D_x} \log \sigma_{t,j}^2)
\end{aligned} \quad (4.11)$$

Note that $q(\mathbf{x}_t|\mathbf{y}_t;\phi) = \mathcal{N}(\mathbf{x}_t;\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)$ and $H(\mathbf{x}_t|\mathbf{y}_t)$ is its entropy. And

$$\begin{aligned} &D_{\mathrm{KL}}(q(\mathbf{x}_t|\mathbf{y}_t;\phi)\|p(\mathbf{x}_t|z_t;\theta)) = \\ &\frac{1}{2}\sum_{j=1}^{D_x}(\frac{\sigma_{t,j}^2}{\sigma_{k,j}^2} + \frac{(\mu_{k,j}-\mu_{t,j})^2}{\sigma_{k,j}^2} - 1 + \log\sigma_{k,j}^2 - \log\sigma_{t,j}^2) \end{aligned} \tag{4.12}$$

such that

$$\begin{aligned} &\mathbb{E}_{q(\mathbf{x}_t|\mathbf{y}_t;\phi)}\left[\log p(\mathbf{x}_t|z_t;\theta)\right] = \\ &-\frac{D_x}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{D_x}(\frac{\sigma_{t,j}^2}{\sigma_{k,j}^2} + \frac{(\mu_{k,j}-\mu_{t,j})^2}{\sigma_{k,j}^2} + \log\sigma_{k,j}^2) \end{aligned} \tag{4.13}$$

Then we can compute Eq. 4.9 accordingly, and apply the Viterbi algorithm to find the most probable HMM state sequence $\{\tilde{z}_t\}_{t=1}^T$, along with the resulting inferred acoustic unit sequence $\mathbf{a} = a_1, ..., a_n$.

VAE-HMM: M-step to maximize the objective function. The objective function is given by the variational lower bound of Eq. 4.2 with the hidden variables $\mathbf{H} = \{\mathbf{X}, \mathbf{Z}\}$:

$$\mathcal{L}(\mathbf{Y};\theta,\gamma,\phi) = \mathbb{E}_{q(\mathbf{X},\mathbf{Z}|\mathbf{Y};\phi)}\left[\log p(\mathbf{Y}|\mathbf{X},\mathbf{Z};\theta,\gamma)\right] - D_{\mathrm{KL}}(q(\mathbf{X},\mathbf{Z}|\mathbf{Y};\phi)\|p(\mathbf{X},\mathbf{Z};\theta)) \tag{4.14}$$

where

$$\begin{aligned} \mathbb{E}_{q(\mathbf{X},\mathbf{Z}|\mathbf{Y};\phi)}\left[\log p(\mathbf{Y}|\mathbf{X},\mathbf{Z};\theta,\gamma)\right] &= \mathbb{E}_{q(\mathbf{X}|\mathbf{Y};\phi)}\left[\log p(\mathbf{Y}|\mathbf{X};\theta,\gamma)\right] \\ &= \sum_{t=1}^{T}\mathbb{E}_{q(\mathbf{x}_t|\mathbf{y}_t;\phi)}\left[\log p(\mathbf{y}_t|\mathbf{x}_t;\gamma)\right] \\ &\simeq \sum_{t=1}^{T}(\frac{1}{L}\sum_{l=1}^{L}\log p(\mathbf{y}_t|\tilde{\mathbf{x}}_t^{(l)};\gamma)) \end{aligned} \tag{4.15}$$

$\tilde{\mathbf{x}}_t^{(l)}$ is drawn from $q(\mathbf{x}_t|\mathbf{y}_t;\phi)$ (Eq. 4.5 and 4.6), and

$$
\begin{aligned}
& D_{\mathrm{KL}}(q(\mathbf{X},\mathbf{Z}|\mathbf{Y};\phi)\|p(\mathbf{X},\mathbf{Z};\theta)) \\
&= \mathbb{E}_{q(\mathbf{Z}|\mathbf{Y};\phi)}\mathbb{E}_{q(\mathbf{X}|\mathbf{Y};\phi)}\left[\log\frac{q(\mathbf{X}|\mathbf{Y};\phi)}{p(\mathbf{X}|\mathbf{Z};\theta)}\right] - \mathbb{E}_{q(\mathbf{Z}|\mathbf{Y};\phi)}\left[\log p(\mathbf{Z};\theta)\right] + const \\
&= \sum_{t=1}^{T}(\mathbb{E}_{\tilde{q}(z_t|\mathbf{Y})}\left[D_{\mathrm{KL}}(q(\mathbf{x}_t|\mathbf{y}_t;\phi)\|p(\mathbf{x}_t|z_t;\theta))\right] - \mathbb{E}_{\tilde{q}(z_{t-1},z_t|\mathbf{Y})}\left[\log p(z_t|z_{t-1};\theta)\right]) \\
&\quad + const
\end{aligned}
\tag{4.16}
$$

We denote $\mathcal{L}(\mathbf{Y};\theta,\gamma,\phi) \simeq \sum_{t=1}^{T}\tilde{\mathcal{L}}(\mathbf{y}_t;\theta,\gamma,\phi)$, and

$$
\begin{aligned}
\tilde{\mathcal{L}}(\mathbf{y}_t;\theta,\gamma,\phi) = \frac{1}{L}\sum_{l=1}^{L}\log p(\mathbf{y}_t|\tilde{\mathbf{x}}_t^{(l)};\gamma) \\
- \mathbb{E}_{\tilde{q}(z_t|\mathbf{Y})}\left[D_{\mathrm{KL}}(q(\mathbf{x}_t|\mathbf{y}_t;\phi)\|p(\mathbf{x}_t|z_t;\theta))\right] \\
+ \mathbb{E}_{\tilde{q}(z_{t-1},z_t|\mathbf{Y})}\left[\log p(z_t|z_{t-1};\theta)\right] \\
+ const
\end{aligned}
\tag{4.17}
$$

We aim to optimize Eq. 4.17 with respect to θ, γ and ϕ.

The first term in Eq. 4.17 is a function of each mean square error (MSE) between the decoder output $f(\tilde{\mathbf{x}}_t^{(l)};\gamma)$ and observation $\mathbf{y}_t$:

$$
\log p(\mathbf{y}_t|\tilde{\mathbf{x}}_t^{(l)};\gamma) = -\frac{\|f(\tilde{\mathbf{x}}_t^{(l)};\gamma) - \mathbf{y}_t\|^2}{2\sigma_y^2} + const \tag{4.18}
$$

which also represents the negative scaled reconstruction loss.

To compute the second and third terms in Eq. 4.17, we first perform Viterbi decoding and find the 1-best sequence $\{\tilde{z}_t\}_{t=1}^{T}$ via the above E-step

inference of Eq. 4.9 – 4.13. Then we use $\{\tilde{z}_t\}_{t=1}^{T}$ to perform Viterbi training and approximate the expectation as:

$$\mathbb{E}_{\tilde{q}(z_t|\mathbf{Y})}\left[D_{\mathrm{KL}}(q(\mathbf{x}_t|\mathbf{y}_t;\phi)\|p(\mathbf{x}_t|z_t;\theta))\right] \simeq D_{\mathrm{KL}}(q(\mathbf{x}_t|\mathbf{y}_t;\phi)\|p(\mathbf{x}_t|\tilde{z}_t;\theta)) \quad (4.19)$$

which is given by Eq. 4.12.

Finally we alternate between the E-step to infer $q(\mathbf{Z}|\mathbf{Y};\theta,\gamma,\phi)$ (Eq. 4.9) and the M-step to maximize the objective function (Eq. 4.17).

4.4.3 Contextual VAE-HMM

In the standard VAE, as shown in the Figure 4.2 and Eq. 4.5 – 4.6, the inference of $q(\mathbf{x}|\mathbf{y}_t;\phi)$ only depends on $\mathbf{y}_t$ regardless of $\mathbf{Y} \setminus \mathbf{y}_t$. We proceed with our investigation on incorporating the additional context frames to better estimate the latent representation and phonetic category of the center frame.

For each time frame t, we use a truncated context window and consider its L/R nearest left/right context frames. Denote the vector concatenation operation as $\oplus$, and the new observation vector $\mathbf{y}'_t$ for each time t is created as $\mathbf{y}'_t = \mathbf{y}_{t-L} \oplus \mathbf{y}_{t-L+1} \oplus \cdots \oplus \mathbf{y}_{t+R-1} \oplus \mathbf{y}_{t+R}$. Therefore, given the new observations $\mathbf{Y'} = \{\mathbf{y}'_t\}_{t=1}^{T}$, we can use the same feedforward NNs as the VAE encoder and decoder networks, and perform the same VAE algorithms as Section 4.4.2.3, referred to as contextual VAE with DNN decoder, as shown in Figure 4.3. The DNN decoder factorizes the joint distribution $p(\mathbf{y}'_t|\tilde{\mathbf{x}}_t)$ as:

$$p(\mathbf{y}'_t|\tilde{\mathbf{x}}_t) = p(\mathbf{y}_{t-L}, \mathbf{y}_{t-L+1}, \ldots, \mathbf{y}_{t+R-1}, \mathbf{y}_{t+R}|\tilde{\mathbf{x}}_t) = \prod_{\tau=t-L}^{t+R} p(\mathbf{y}_\tau|\tilde{\mathbf{x}}_t) \quad (4.20)$$

where each $\mathbf{y}_\tau$ is assumed to be independent of each other and is conditioned

only on $\tilde{\mathbf{x}}_t$, such that the VAE model has to encode all the information of $\mathbf{y}'_t$ into a single vector $\tilde{\mathbf{x}}_t$ to reconstruct $\mathbf{y}'_t$.

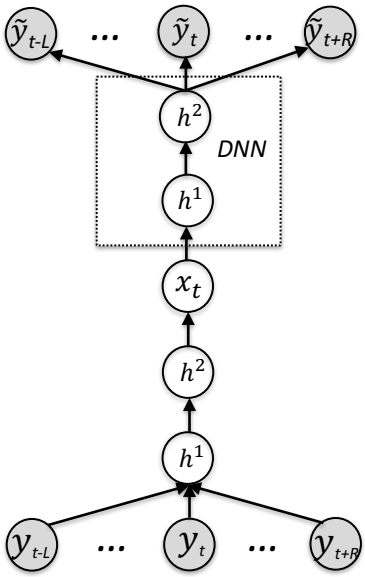


Figure 4.3: Contextual VAE with 2-hidden layer DNN encoder and decoder.

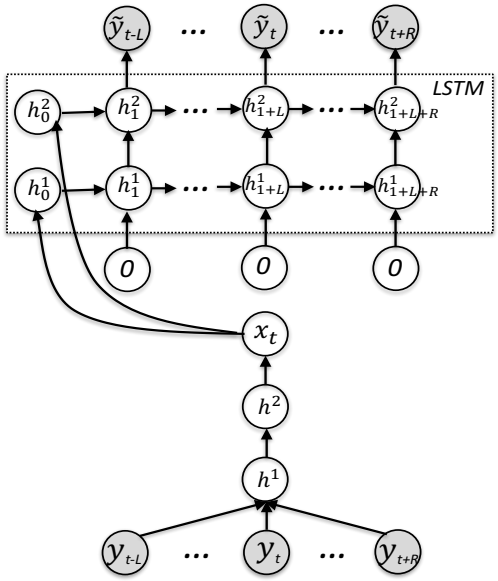


Figure 4.4: Contextual VAE with 2-layer LSTM decoder.

Additionally, we can also use an LSTM instead of DNN as the decoder network in contextual VAE. As described in [115], the LSTM decoder can factorize $p(\mathbf{y}'_t|\tilde{\mathbf{x}}_t)$ with the chain rule:

$$\begin{aligned} p(\mathbf{y}'_t|\tilde{\mathbf{x}}_t) &= p(\mathbf{y}_{t-L}, \mathbf{y}_{t-L+1}, \cdots, \mathbf{y}_{t+R-1}, \mathbf{y}_{t+R}|\tilde{\mathbf{x}}_t) \\ &= p(\mathbf{y}_{t-L}|\tilde{\mathbf{x}}_t) \prod_{\tau=t-L+1}^{t+R} p(\mathbf{y}_\tau|\mathbf{y}_{\tau-1}, \cdots, \mathbf{y}_{t-L}, \tilde{\mathbf{x}}_t) \end{aligned} \tag{4.21}$$

Thus, to reconstruct $\mathbf{y}'_t$ it allows for capturing the sequential dependencies across $\{\mathbf{y}_\tau\}_{\tau=t-L}^{t+R}$, which relieves the model from encoding every single detail in the sequence $\{\mathbf{y}_\tau\}_{\tau=t-L}^{t+R}$.

Specifically, we first draw $\tilde{\mathbf{x}}_t$ from $q(\mathbf{x}_t|\mathbf{y}'_t; \phi)$, and use $\tilde{\mathbf{x}}_t$ to predict (via an

affine transformation) the initial hidden states of the decoder LSTM (but not the cell states); after initialization, the decoder LSTM network takes a zero vector as input at each time step, and generates a sequence of outputs. Then each output goes through an affine transformation to predict the mean of $\mathbf{y}_\tau$, for each $\tau = t - L, \ldots, t + R$. The process is illustrated in Figure 4.4. Note that we use the historyless decoding technique, i.e. zero vectors as each input to the decoder LSTM, inspired by [115, 116], such that the decoder is forced to ignore the history and relies fully on the latent representation $\tilde{\mathbf{x}}_t$.

4.4.4 Experiments

4.4.4.1 Evaluation metric

To evaluate the quality of the automatically learned acoustic models, we compute the normalized mutual information (NMI) between the hypothesized acoustic unit sequences and the orthographic phoneme transcripts. We first obtain acoustic unit tokenizations, i.e., 1-best HMM unit-level decode, of the development (dev) data on which AUD training is performed; alternatively, we can also use the learned models to obtain tokenizations of any evaluation data that the models do not see during training. Then we align the decoded acoustic unit sequence $\boldsymbol{a} = a_1, ..., a_n$ with reference phoneme sequence $\boldsymbol{p} = p_1, ..., p_m$, and thus each $a_j (1 \leq j \leq n)$ is aligned to a $p_i (1 \leq i \leq m)$, based on which the mutual information $I(\mathbf{p}; \mathbf{a})$ is computed. We normalize it by the entropy $H(\mathbf{p})$ of $\mathbf{p}$, giving the normalized mutual information:

$$NMI(\mathbf{p}; \mathbf{a}) = \frac{I(\mathbf{p}; \mathbf{a})}{H(\mathbf{p})} \tag{4.22}$$

where $NMI(\mathbf{p};\mathbf{a}) = 0$ means $\mathbf{a}$ carries no information about $\mathbf{p}$, and $NMI(\mathbf{p};\mathbf{a}) = 1$ means $\mathbf{a}$ perfectly predicts $\mathbf{p}$.

4.4.4.2 Datasets

We evaluate our AUD performance on two corpora. First, we perform AUD on the TIMIT [117] training corpus (∼3.9 hrs), obtain the acoustic unit tokenizations, and compute NMI on the TIMIT test corpus (∼1.4 hrs). The number of distinct reference phonemes on TIMIT is 61.

Second, we perform AUD on Switchboard Telephone Speech Corpus [118], a collection of two-sided telephone conversations. Following [54, 93], we use the same development (dev, 35.7 hrs) and evaluation (eval, 61.6 hrs) datasets[4]. We use manual segmentations provided by the Switchboard corpus to produce utterances with speech activity, which AUD further operates on. We perform AUD training only on dev set, and compute NMI on both dev and eval sets. The number of distinct reference phonemes is 46.

4.4.4.3 Acoustic feature representations

For TIMIT, we parameterize it into 39-dimensional MFCCs with first and second order derivatives.

For Switchboard, we conduct our multilingual bottleneck (BN) network training. We use the time delay neural network (TDNN) [74] with two major modifications. First, hidden layers with rectified linear unit (ReLU) nonlinearity are shared across languages, where 10 language collections[5] from IARPA

[4]More details will be described in Section 4.6.1.1.

[5]Assamese, Bengali, Cantonese, Haitian, Lao, Pashto, Tamil, Tagalog, Vietnamese and

Babel Program [10] and about 10-hour transcribed speech of each language are used. 40-dimensional MFCCs (without cepstral truncation [74]) augmented with 3-dimensional pitch and probability of voicing features are used as inputs to the network. The final output layer is a set of individual language-specific output layers with context-dependent triphone state targets. Second, an additional 42-dimensional bottleneck layer is added just before the final output layer, so the final BN features are 42-dimensional. The complete architecture is illustrated in Figure 4.5.

Finally we apply Cepstral mean and variance normalization (CMVN) per utterance to the MFCCs of TIMIT, and CMVN per conversation side to the BN features of Switchboard, on both of which AUD further operates.

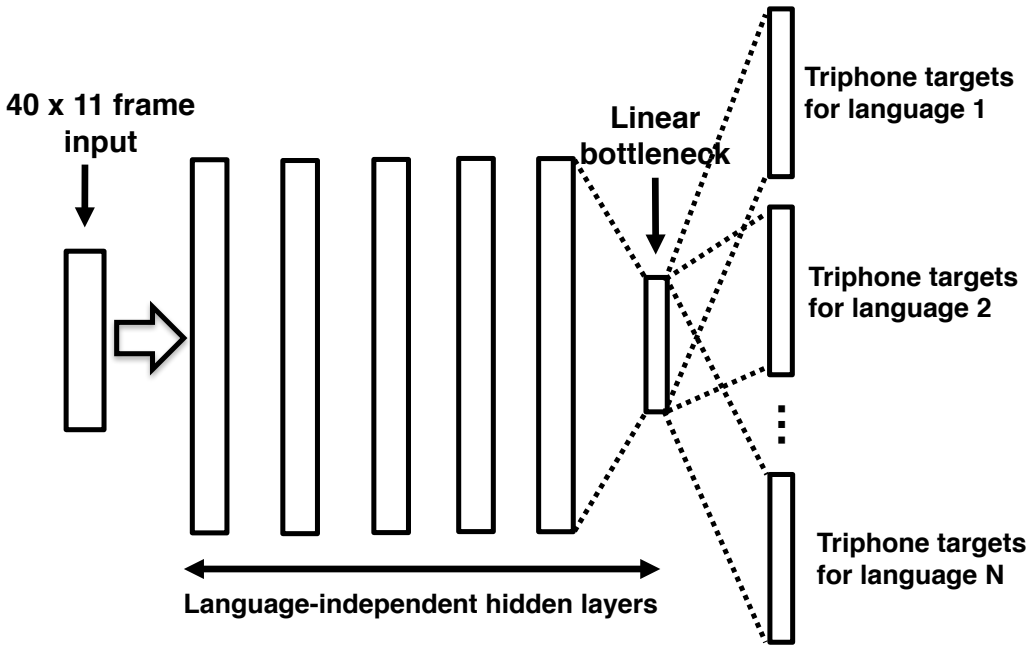


Figure 4.5: The configuration of our multilingual TDNN-based bottleneck network.

Zulu.

4.4.4.4 Model configurations

GMM-HMM. We use the truncation level $M = 200$, which implies maximum 200 different acoustic units can be learned from each corpus. Each acoustic unit is modeled by 3-state HMM with a left-to-right topology and 2 Gaussians per state. For the stick-breaking construction of Dirichlet process, we use concentration parameter $\gamma = 1.0$ on TIMIT, and $\gamma = 10.0$ on Switchboard. Unsupervised AUD training is stopped after 10 epochs. Other hyperparameter values are the same as [106].

VAE-HMM. After we use the GMM-HMM system to do Viterbi decoding and obtain an HMM state sequence, we use such state-level sequence as the $\{\tilde{z}_t\}_{t=1}^{T}$ in Eq. 4.19 for pre-training the VAE-HMM. After pre-training (3 epochs), the subsequent unsupervised learning proceeds as described in Section 4.4.2.3. Note that the count U of acoustic units is determined from the GMM-HMM system, and here we do not continue to update the mixture weights $\{\pi_m\}_{m=1}^{U}$. Training is stopped with a fixed number of epochs or a minimal change of the training objective on a small validation set. The encoder and decoder networks are feedforward NNs of 2 ReLU layers with 256 hidden units. The latent representation $\mathbf{x}$ is 32-dimensional.

Contextual VAE-HMM. We experiment with the context window size from $L = R = 1$ to $L = R = 5$ (Section 4.4.3). The DNN encoder and decoder networks are feedforward NNs of 2 ReLU layers with 512 hidden units. The LSTM decoder network is 2-layer with 512-dimensional hidden states where

each zero input is 96-dimensional. The latent space of $\mathbf{x}$ is 96-dimensional.

Table 4.1: Infinite HMM based AUD performance on TIMIT using MFCCs.

Acoustic Model		NMI
GMM		39.09
VAE		41.17
Contextual VAE	DNN decoder	44.25
	LSTM decoder	44.42

Table 4.2: Infinite HMM based AUD performance on Switchboard using multilingual bottleneck features.

Dataset	Acoustic Model		NMI
Dev	GMM		29.61
	VAE		34.49
	Contextual VAE	DNN decoder	35.25
		LSTM decoder	35.52
Eval	GMM		29.11
	VAE		33.90
	Contextual VAE	DNN decoder	34.65
		LSTM decoder	34.92

4.4.4.5 Results and discussion

The NMI results on TIMIT are shown in Table 4.1. The number of distinct discovered units on training corpus is 112. The VAE-HMM, which combines the strengths of deep learning and probabilistic graphical models, significantly outperforms the GMM-HMM baseline. The contextual VAE-HMM alternative gives the best results by using context window size $L = R = 4$, where the LSTM decoder slightly outperforms the DNN decoder. Overall, the contextual VAE-HMM produces large gains over the VAE-HMM.

We find similar results on Switchboard as shown in Table 4.2, and the number of distinct discovered units on dev set as 199. However, note that to produce the BN feature for each time frame, its left and right context frames have been stacked as inputs to the BN network, so that the BN feature of each center frame has been a context-dependent acoustic representation. This indicates why the contextual VAE-HMM on the BN features of Switchboard does not produce as large a gain over VAE-HMM as on the context-independent MFCCs of TIMIT. Also, although the unsupervised AUD training is only performed on Switchboard dev set, we see little NMI degradation between dev and eval sets, and it shows that a relatively robust generalization of AUD models to the unseen data.

Above all, we propose a high-performing contextual VAE-HMM based AUD framework. First, it is able to automatically learn subword units that are highly correlated with orthographic phonemes. Second, it segments speech into sequences of phoneme-like units, and gives an effective approach to obtaining speech tokenizations that can be used to create spoken document representations, which we discuss below.

4.5 Document representation and classification

We use document representation to refer to either a vector representation for a spoken document, or a vector representation for each speech segment if the document is segmented into a sequence of segments. This chapter focuses on learning representation and performing classification either for each spoken document, or for each speech segment independently, in the absence of the

across-segment contextual modeling effects introduced in the previous Section 3.4.3 and 3.4.4.

4.5.1 Bag-of-words representation

After we obtain the tokenizations of speech by UTD or AUD, each spoken document/segment is represented by a vector of unigram occurrence counts over discovered terms, or a vector of n-gram counts over acoustic units, respectively. Similarly in Section 3.4.1, each vector can be further scaled to produce a term frequency-inverse document frequency (tf-idf) feature.

Also as described in Section 3.4.2, given the bag-of-words representation, we use a stochastic gradient descent (SGD) based linear SVM [78, 79] with hinge loss and $\mathcal{L}^1/\mathcal{L}^2$ norm regularization for single-label classification. In the setting where each spoken document/segment is associated with multiple labels, we proceed to perform a multi-label classification task. The baseline approach is the binary relevance method, which independently trains one binary classifier for each label, and a test data instance is evaluated by each classifier to determine if the respective label applies to it. Specifically, we use a set of SVMs, one for each label.

4.5.2 Convolutional neural network-based representation and classification

AUD enables full-coverage tokenization of continuous speech into a sequence of acoustic units, which we can exploit in a CNN-based framework to learn a vector representation for each spoken document/segment. As shown in

Figure 4.6, in an acoustic unit sequence **a** of length m, each unit a_i, $1 \leq i \leq m$, is encoded as a fixed dimensional continuous vector, and the whole sequence **a** is represented as a concatenated vector **x**. A shared convolutional feature transform T spans a fixed-sized n-gram window, $n \ll m$, and slides over the whole sequence. Then the hidden feature layer $\mathbf{h}^1$ with nonlinearities consists of each feature vector h_i^1 extracted from the shared convolutional window centered at each acoustic unit position i. Max-pooling is performed on top of each h_i^1, $1 \leq i \leq m$, to obtain a fixed-dimensional vector representation for the whole sequence **a**, i.e., a vector representation of the whole spoken document, followed by another hidden layer $\mathbf{h}^2$ and a final output layer.

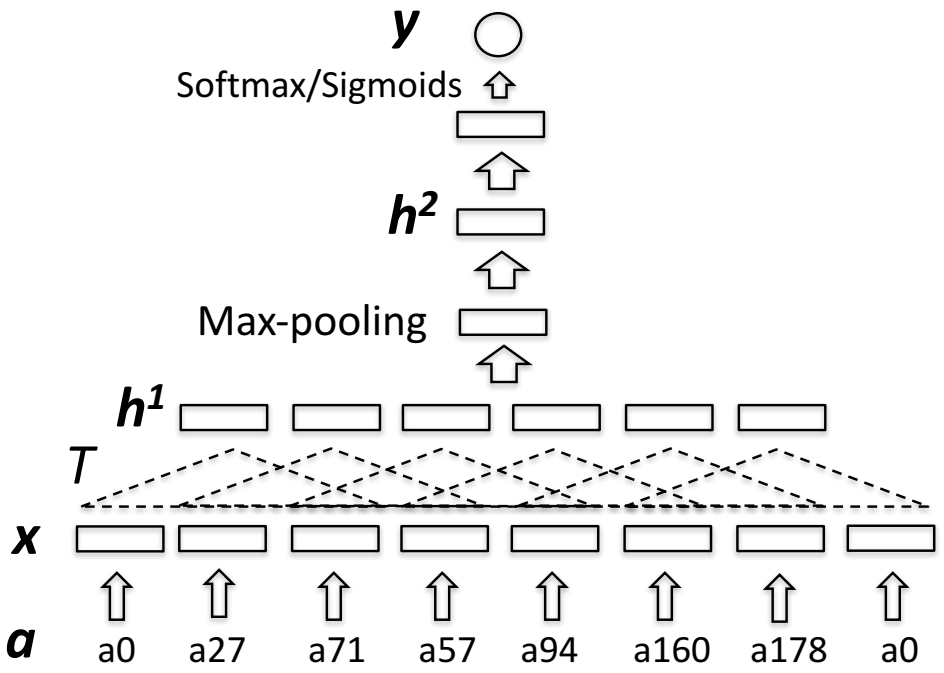


Figure 4.6: CNN-based framework that operates on automatically discovered acoustic units.

Note that this framework needs supervision for training. The output layer is a softmax function for single-label classification, and the whole model is

trained with categorical cross-entropy loss. For multi-label classification, as similarly in Section 3.4.2, we replace the softmax in the output layer with a set of sigmoid output nodes, one for each label. Since a sigmoid naturally provides output values between 0 and 1, we train the network to minimize the binary cross entropy loss defined as

$$l(\Theta; \mathbf{x}, \mathbf{y}) = -\sum_{k=1}^{K} (y_k \log o_k + (1 - y_k) \log(1 - o_k)) \tag{4.23}$$

where Θ denotes the CNN parameters, $\mathbf{x}$ is the concatenated feature vector of acoustic unit sequence, $\mathbf{y}$ is the target binary vector of labels, o_k and y_k are the output and the target for label k, and the number of unique labels is K.

Also, the vector representation of each unique acoustic unit can be randomly initialized, or pre-trained from other tasks. Specifically, we apply the *skip-gram* model of word2vec [119] to pre-train one embedding vector for each acoustic unit, via the hierarchical softmax with Huffman codes.

4.6 Experiments

4.6.1 Single-label classification

4.6.1.1 Experimental setup

For our single-label classification experiments, we use the Switchboard Corpus [118], a speech collection of two-sided telephone conversations. We use the same dev and eval data sets as in [54, 93]. Each whole conversation has two sides and one single topic, and classification is performed on each individual-side speech (i.e., each side is seen as one single spoken document). In the 35.7

hour dev data, there are 360 conversation sides evenly distributed across six different topics (recycling, capital punishment, drug testing, family finance, job benefits, car buying), i.e., each topic label has equal number of 60 sides. In the 61.6 hour eval data, there are another different six topics (family life, news media, public education, exercise/fitness, pets, taxes) evenly distributed across 600 conversation sides. Algorithm design choices are explored through experiments on dev data. We use manual segmentations provided by the Switchboard corpus to produce utterances with speech activity, and use the same multilingual bottleneck features as in Section 4.4.4.3, which UTD and AUD further operate on.

For UTD, we use the ZRTools [100] implementation as described in Section 4.3, with the default parameters except that, we use cosine similarity threshold $\delta = 0.5$, and vary the diagonal median filter duration κ over $\{0.6, 0.7\}$; we try both the exact DTW-based similarity and the rescored similarity, and tune the similarity threshold (used to partition the graph edges) over $\{0.85, 0.88, 0.90, 0.92\}$.

For AUD, we experiment with both the GMM-HMM (Section 4.4.1) and VAE-HMM (Section 4.4.2.3) based models[6], with the same configurations as described in 4.4.4.4; except for the stick-breaking construction of Dirichlet process, we vary the concentration parameter γ over $\{1.0, 10.0\}$.

For SVM-based classification, we use the bag of discovered term unigrams, or bag of acoustic unit trigrams. On dev data, we try using the features of

[6]We do not employ contextual VAE-HMM (Section 4.4.3) here, since we observe few gains when it operates on multilingual bottleneck features, as discussed in Section 4.4.4.5.

raw counts or the features scaled by inverse document frequency. SVM regularization is tuned over $\mathcal{L}^1/\mathcal{L}^2$ norm, regularization constant tuned over $\{0.001, 0.0001\}$, and SGD epochs tuned over $\{30, 50\}$. We further normalize each feature to $\mathcal{L}^2$ norm unit length. Each experiment is a run of 10-fold cross validation (CV) on the 360 conversation sides of dev data, or on the 600 sides of eval data, respectively. Note that our data size here is relatively small (only 360 or 600) and the SGD training may give high variance in the performance [120]. Therefore, to report classification accuracy for each configuration (when varying features or models), we repeat each CV experiment 5 times, where each experiment again is a run of 10-fold CV; then for each configuration, the mean and standard deviation of 5 experiments is reported.

For CNN-based classification, we use the same strategy to report classification accuracy, i.e., repeating experiments 5 times (where each time is a 10-fold CV) for each CNN configuration. Note that the respective 10 folds of both dev and eval data sets are fixed the same for all the SVM and CNN experiments. Additionally, for each 10-fold CV experiment, instead of training on 9 folds and testing on the remaining 1 fold as in SVM, we use 8 folds for CNN training, leave another 1 fold out as validation data; after training each CNN model for up to 100 epochs, the model with the best accuracy on the validation data is used for evaluation on the test set. The acoustic unit sequence (as CNN inputs) are zero-padded to the longest length in each dataset. We implemented the CNNs in Keras [121] with Theano [122] backend. CNN architectures are determined through experiments on dev data. For SGD training we use the Adadelta optimizer [123] and mini-batch size

18. The n-gram window size of each convolutional feature transform T is 7. The size of each hidden feature vector h_i^1 (extracted from the transform T) is 1024, with ReLU nonlinearities. Thus, after max-pooling over time, we have a 1024-dimensional vector again, which then goes through another hidden layer $\mathbf{h}^2$ (also set as 1024-dimensional with ReLU) and finally into a softmax. Dropout [124] rate 0.2 is used at each layer.

When we initialize the vector representation of each acoustic unit with a set of pre-trained vectors (instead of random initializations), we apply the skip-gram model of word2vec [119] to the acoustic unit tokenizations of each data set. We use the *gensim* implementation [125], which includes a vector space of embedding dimension 50 (tuned over $\{50, 80\}$), a skip-gram window of size 5, and SGD over 20 epochs.

4.6.1.2 Results on Switchboard

Table 4.3 shows the document classification results on Switchboard. For UTD-based classifications, we find that the default rescoring in ZRTools [100], which is designed to filter out the filled pauses, produces comparable performance to the raw DTW similarity scores, but the rescoring can result in much faster connected-component clustering (Section 4.3). Note that this rescoring model is estimated using a portion of transcribed Switchboard, but it is still a legitimate language-independent UTD approach while operating on languages other than English. While a diagonal median filter duration κ of 0.6 or 0.7 gives similar results, $\kappa = 0.7$ produces longer but fewer terms, giving more sparse feature representations. Therefore, we proceed with rescoring and

$\kappa = 0.7$ for the following UTD experiments in Section 4.6.2.

Table 4.3: Single-label classification accuracies on Switchboard.

Dataset	Feature	Config	Topic Model	Accuracy
Dev	UTD	–	SVM	0.863 ± 0.010
	UTD	rescoring	SVM	$\mathbf{0.876} \pm 0.008$
	GMM-HMM AUD	# units 184	SVM	0.682 ± 0.007
			CNN	0.657 ± 0.017
			CNN w/ word2vec	0.728 ± 0.011
	GMM-HMM AUD	# units 199	SVM	0.686 ± 0.005
			CNN	0.749 ± 0.008
			CNN w/ word2vec	0.763 ± 0.011
	VAE-HMM AUD	# units 199	SVM	0.730 ± 0.006
			CNN w/ word2vec	$\mathbf{0.793} \pm 0.010$
Eval	UTD	–	SVM	0.851 ± 0.003
	UTD	rescoring	SVM	$\mathbf{0.875} \pm 0.003$
	GMM-HMM AUD	# units 184	SVM	0.710 ± 0.005
			CNN	0.708 ± 0.013
			CNN w/ word2vec	0.762 ± 0.007
	GMM-HMM AUD	# units 199	SVM	0.700 ± 0.005
			CNN	0.690 ± 0.015
			CNN w/ word2vec	0.767 ± 0.013
	VAE-HMM AUD	# units 199	SVM	0.777 ± 0.003
			CNN w/ word2vec	$\mathbf{0.823} \pm 0.005$

For the classifications that use the units from GMM-HMM based AUD, CNN without word2vec pre-training usually gives comparable results with SVM; however, using word2vec pre-training, CNN substantially outperforms the competing SVM in all cases. Also as the concentration parameter γ in AUD increases from 1.0 to 10.0 (yielding less concentrated distributions), we have more unique acoustic units in the tokenizations of both data sets, from 184 to 199, and $\gamma = 10.0$ usually produces better results than $\gamma = 1.0$.

Also, the results based on VAE-HMM AUD are dramatically better than

those based on GMM-HMM AUD, and such classification performance gains are consistent with the NMI improvements shown in Section 4.4.4.5. Thus, the progress in the intrinsic NMI measure of our AUD model development is demonstrated to predict the improved efficacy of our AUD-based real speech applications.

4.6.2 Multi-label classification

4.6.2.1 Experimental setup

We further evaluate our classification performance on the same speech corpora released by the DARPA LORELEI (Low Resource Languages for Emergent Incidents) Program, as introduced in Section 3.5.1.1 of Chapter 3. For each language there are a number of speech segments, and each speech segment is viewed as either in-domain or out-of-domain. In-domain data is defined as any speech segment relating to an incident or incidents, and in-domain data will fall into a set of domain-specific topic categories; these categories are known as situation types, or in-domain topics, as shown in Table 3.1 of Chapter 3. There are 11 situation types: "Civil Unrest or Wide-spread Crime", "Elections and Politics", "Evacuation", "Food Supply", "Urgent Rescue", "Utilities, Energy, or Sanitation", "Infrastructure", "Medical Assistance", "Shelter", "Terrorism or other Extreme Violence", and "Water Supply". We consider "Out-of-domain" as the 12th topic label, so each speech segment either corresponds to one or multiple in-domain topics, or is "Out-of-domain".

In this chapter, classification is always performed on each speech segment independently. As similarly in Section 3.5.1.2, we use average precision (AP,

equal to the area under the precision-recall curve) as the evaluation metric. However, we not only compute the micro-averaged precisions and recalls across 11 in-domain topic labels, but also compute them across the overall 12 labels (including the "Out-of-domain" label). Thus, we report both the AP across 11 in-domain topics, and the AP across overall 12 labels, as the evaluation results.

For each configuration, only a single 10-fold CV result is reported, since we observe less variance in results here than in Switchboard. We have 16.5 hours in-domain data and 8.5 hours out-of-domain data for Turkish, 7.7 and 7.2 hours for Mandarin, and the splits of rest three languages, Tigrinya, Oromo and Russian, are shown in Figure 4.7, 4.8 and 4.9. We use the same CNN architecture as on Switchboard but make the changes as described in Section 4.5.2. Also we use mini-batch size 30 and fix the training epochs as 100. All CNNs use word2vec pre-training.

Additionally, we implement another set of classification baselines using the standard ASR systems built with transcribed speech. Turkish ASR is trained with 80 hour transcribed Turkish telephone conversational speech from Babel corpus [10]. One Mandarin ASR is trained with about 170 hour transcribed HKUST Mandarin telephone speech (LDC2005T32 and LDC2005S15), and the other is trained with about 600 hour GALE Chinese Broadcast News Speech [84]. The acoustic models are the sequence-trained TDNNs based on lattice-free maximum mutual information (LF-MMI) [75]. Note that most LORELEI speech is broadcast news, so there is severe domain and channel mismatch between LORELEI datasets and the ASR systems built with telephone speech.

Thus we also experiment with including the monolingual text provided by the LORELEI language packs into the language model training data, and rebuilding the decoding graph with the new language model, such that the vocabulary size is expanded to mitigate the domain mismatch issue.

Also, for Tigrinya, Oromo and Russian of which transcribed speech is unavailable to us, we employ the universal phone set ASR introduced in Section 3.3 and 3.5.1.3 of Chapter 3 as another baseline.

Again, the acoustic features on which UTD and AUD operate are multilingual bottleneck features as described in Section 4.4.4.3, while here we conduct the multilingual BN network training with 24 Babel language collections[7], and about 10 hours per language.

4.6.2.2 Results on LORELEI datasets

As shown in Table 4.4, we note that on LORELEI datasets, UTD-based systems do not always outperform AUD-based ones as what we find on Switchboard (Section 4.6.1.2), presumably because LORELEI speech is much more noisy and as compared to the model-based AUD, the frame-wise cosine similarity computations in DTW-based UTD are less robust to noisy speech frames.

Note that CNN-based systems dramatically outperform SVMs on the larger sized Switchboard datasets (35.7/61.6 hours, Section 4.6.1), while the CNNs on LORELEI corpora do not produce as a gain over SVMs as on Switchboard. Since each 15-25 hour LORELEI corpus with 12 topic labels is a relatively

[7]Cantonese, Assamese, Bengali, Pashto, Turkish, Tagalog, Vietnamese, Haitian, Swahili, Lao, Tamil, Kurmanji, Zulu, Tokpisin, Cebuano, Kazakh, Telugu, Guarani, Igbo, Amharic, Mongolian, Javanese, Dholuo and Georgian.

Table 4.4: Multi-label classification average precisions on two LORELEI languages. Vocab expansion denotes the use of a new language model that includes additional monolingual text during training. 'In-domain' denotes the ASR built with about 600 hour transcribed Chinese broadcast news speech.

Dataset	Feature	Model	Overall	In-domain topics
Turkish (24.96 hours, 2095 segments)	UTD	SVM	0.627	0.577
	AUD	SVM	0.672	0.614
	AUD	CNN	0.673	0.608
	ASR	SVM	0.644	0.598
	ASR, vocab expansion	SVM	0.707	0.672
Mandarin (14.89 hours, 724 segments)	UTD	SVM	0.478	0.277
	AUD	SVM	0.469	0.232
	AUD	CNN	0.463	0.231
	ASR	SVM	0.568	0.410
	ASR, vocab expansion	SVM	0.602	0.464
	ASR, in-domain	SVM	0.677	0.558

small amount of data compared to the 35.7/61.6 hour Switchboard corpus with 6 labels, it indicates more supervised labeled data is needed to enable competitive CNNs.

Furthermore, UTD/AUD-based systems achieve comparable results with a domain mismatched Turkish ASR, while falling short on Mandarin. Both ASR systems with vocabulary expansion (i.e. more in-domain language model training data) show substantial improvements, and the in-domain ASR trained with sufficient supervised data gives the topline results.

For Tigrinya, Oromo and Russian where no sufficient transcribed training data is available to build a standard ASR, we employ the universal phone set ASR without adaptation or with adaptation on very small amount of data (Section 3.5.1.3; we use around 10 hour transcribed read speech from VoxForge corpus [126] as the Russian adaptation data). We compare the performance

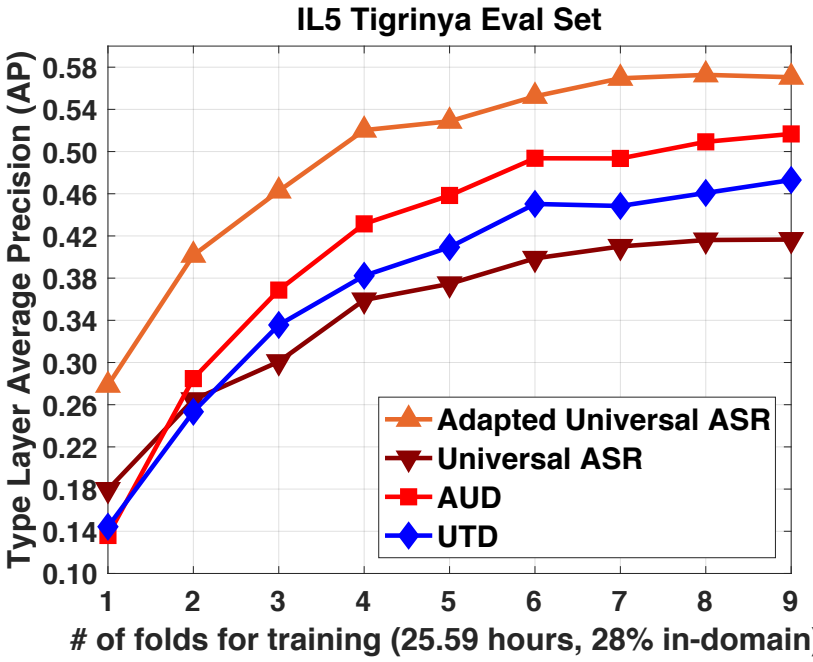


Figure 4.7: 10-fold CV APs on Tigrinya when varying the number of training folds.

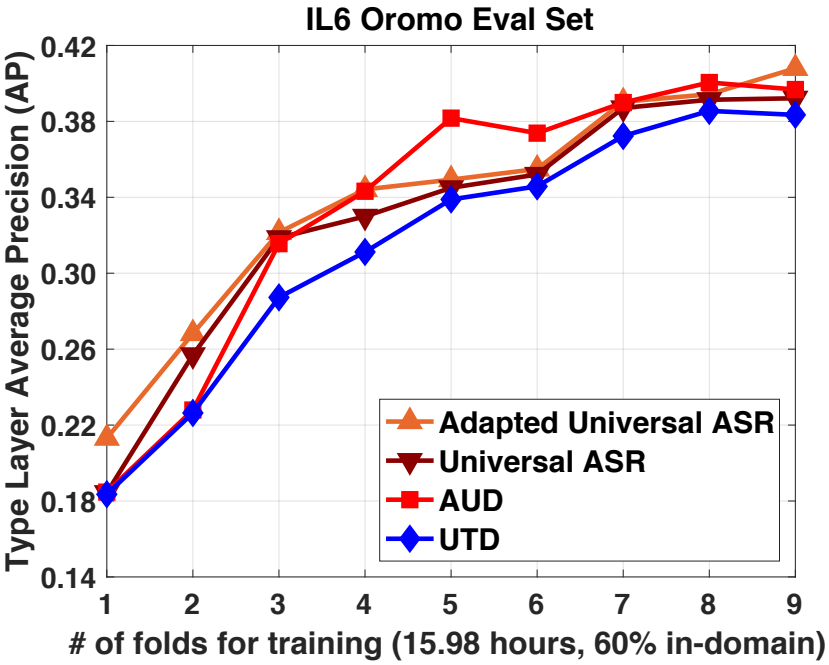


Figure 4.8: 10-fold CV APs on Oromo when varying the number of training folds.

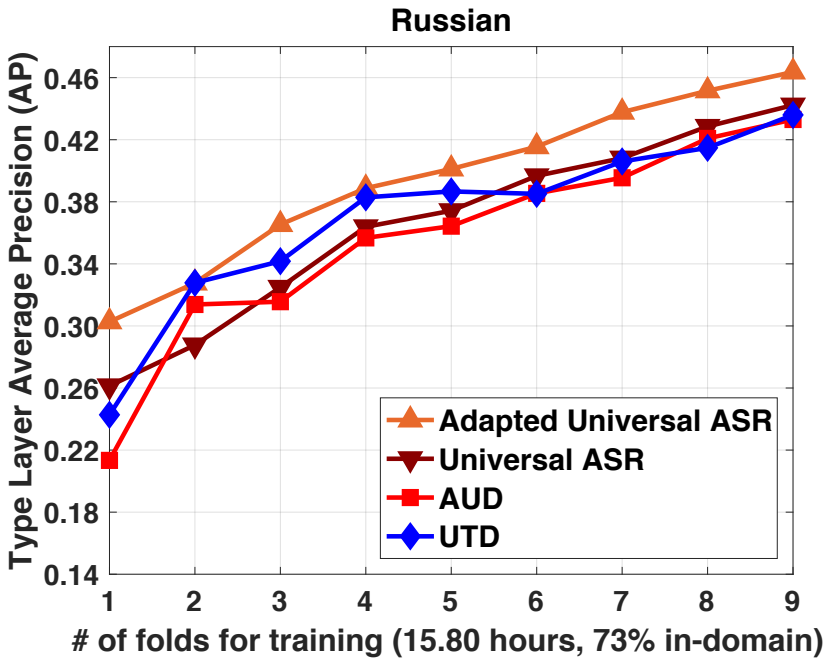


Figure 4.9: 10-fold CV APs on Russian when varying the number of training folds.

among UTD, AUD and ASR on each individual language by varying the amount of training data; we split each dataset into 10 folds, and perform 10-fold CV 9 times (with SVMs), varying the number of folds for training from 1 to 9. As illustrated in Figure 4.7, 4.8 and 4.9, as we use more folds for training, performance improves across the board. Adapted ASR-based systems still give the best results in most cases, while UTD and AUD based ones achieve comparable numbers.

Furthermore, we find in Table 4.4, AUD-based SVMs are more competitive than UTD-based SVMs on the larger corpus, i.e. Turkish, while being less competitive on the smaller sized Mandarin. We also find AUD more competitive on the larger sized Tigrinya in Figure 4.7, while being comparable on smaller sized Oromo and Russian in Figure 4.8 and Figure 4.9.

4.7 Conclusion

We first present a state-of-the-art phonetic discovery approach using contextual VAE-HMM. Then we demonstrate that both UTD and AUD are viable technologies for producing effective tokenizations of speech that enable spoken document classification performance comparable to using a domain-mismatched ASR or a universal phone set ASR. Importantly, such unsupervised approaches remove dependency on the typical linguistic resources that standard ASR alternative strongly relies on.

We find that the classifications with DTW-based UTD outperform the performance with VAE-HMM based AUD on the cleaner Switchboard corpus, while generally falling behind on the more noisy LORELEI corpora. Moreover, given sufficient training data on Switchboard, AUD-based CNNs with word2vec pre-training outperform AUD-based SVMs.

Chapter 5

Conclusion

The body of work contained in this dissertation records the many significant improvements to various speech retrieval techniques which enable the evolution from proof-of-concept experiments on clean and extensively annotated corpora into low-resource efficient systems operated on real-world unstructured speech. Current state-of-the-art speech retrieval technologies strongly rely on various annotated linguistic resources, e.g. transcribed speech and pronunciation lexicons. Given the vast language diversities, sourcing such annotated collections can be difficult or restricted, especially for large quantities of unwritten languages without orthography. Also, the massive volumes of streaming data from sites like YouTube present large challenges to the scalability of the automated processing systems. Therefore, the aim of our work has been to identify effective ways to advance scalable speech retrieval techniques in resource-scarce scenarios.

5.1 Summary

This thesis has been focused on two lines of research, spoken keyword retrieval and spoken document classification.

5.1.1 Low-resource efficient keyword retrieval

The central theme of the first research area – keyword retrieval – is to address the technological challenges necessary to extend the point process model for keyword search in the low-resource settings where the amount of transcribed speech is severely limited and the pronunciation dictionaries are incomplete.

In Chapter 2, we began with introducing the context-dependent DNNs in the context of an LVCSR system, and proceeded to translate the improved DNN acoustic models into more accurate phone posterior estimations, so as to replace the old-fashioned phoneme recognizer that use monophone classes as training targets. In turn, the more accurate phonetic event estimations given to the PPM framework were demonstrated to make for state-of-the-art OOV search performance; also, though the PPM overall performance trailed the HMM-based search, they combined to post dramatic fusion gains over the LVCSR alone.

Furthermore, in order to capture the acoustic variations in differing phonetic context, DNN-HMM based LVCSR is built with the triphone HMMs, and the DNN serves to estimate the tied triphone state (known as senone) posteriors. Also to that end, we aim to enable PPM's compatibility with such context-dependent phonetic modeling. First, we defined the new phonetic event as each tied triphone state event, and extract the event streams from the

same DNN output posteriors as in LVCSR. Second, we developed a procedure to build PPMs based on the tied triphone state labels (instead of the dictionary phonemes used as before), which allowed PPM to decode on the new context-dependent event streams. Experiments of the PPM modeling on such new search index have demonstrated substantially improved search performance.

Finally, we developed a PPM-based lattice generation algorithm. We first ran parallel word detectors for entire vocabulary and used the independent detections to construct a "words-on-nodes" lattice that accommodated the duration uncertainties in PPM decoding. Then we converted it to a standard lattice with word and PPM acoustic likelihood (as acoustic score) on each arc, and processed it with standard FST-based algorithms such as language model composition, KWS indexing and ASR decoding. We showed that such detection-based lattice generation framework provided competitive keyword search and ASR performance, and compared with HMM-based ASR, it is still a computationally light model and being an alternative path to LVCSR.

5.1.2 Spoken document classification for almost-zero-resource languages

The theme of the second research area is to perform spoken document classification for languages where the resources of transcribed speech are scarce.

To transform audio into indexable tokens, we first employed a universal phone set ASR which used a common phonemic representation shared across languages in Chapter 3. After decoding speech into orthographic words, we translated each word into English by looking up a bilingual lexicon that

was either preexisting or derived from the word alignments via a machine translation system; we did not apply a MT system to explicitly decode the speech transcripts, so as to simulate the realistic setting where no parallel training data exists to build the standard MT system. Thus, we were able to build an English-language topic classifier by obtaining English text/topic pairs from multiple resourceful languages, which allows for a near language-agnostic operation. We showed that our systems achieved very competitive results in the NIST LoReHLT 2017 Evaluations [16].

Note that audio collected in the wild can be extremely long, of variable length, and contain multiple class label shifts (e.g. topic shifts) at variable locations in the audio, so each audio instance, known as a spoken document, often needs to be split into a sequence of speech segments. Our above system proceeded to classify each segment individually. However, in Chapter 3 we further outlined novel contextual modeling frameworks that encoded context dependencies across adjacent segments into the classification process. We demonstrated the progression of models from context-independent to context-aware provided considerable performance improvements. Also, our proposed attention based contextual classifiers, which were able to selectively detect and use relevant contexts over irrelevant ones, consistently outperformed the recurrent neural network based alternatives.

In Chapter 4, other than supervised training of a universal phone model as above, we began our investigation of transforming audio into indexable tokens via unsupervised alternatives. The first examined approach was to automatically detect indexable terms via acoustic repetitions, referred to as

UTD. The second exploited approach was to jointly identify a phonemic inventory and segment speech into sequences of phoneme-like units, known as AUD. We demonstrated a proposed context-sensitive variational autoencoder composed with HMMs to achieve the state-of-the-art AUD performance in the intrinsic normalized mutual information measures. To further quantify the effect of our improved AUD models in creating document representations, we proceeded with topic classification experiments on Switchboard datasets. We observed that the classification performance progressed consistently with the NMI improvements, and the proposed CNN based representation learned on the acoustic unit sequences significantly outperformed the bag-of-words representation. Next, we found that, the the classifications via VAE-HMM based AUD trailed the DTW-based UTD results on the cleaner Switchboard, while generally being more competitive on the more noisy LORELEI speech corpora. Unquestionably, the standard ASR systems trained via hundred hours of transcribed speech still gave the topline results. However, we have observed that UTD and AUD based classifications achieved comparable results against the universal phone set ASR. Importantly, the viable unsupervised speech technologies – lexical or phonetic discovery – are able to automatically identify indexing tokens regardless of the language orthography that standard ASR strongly relies on.

5.2 Future directions

We outline a number of promising future directions on further improving the various techniques described in the preceding chapters.

Supporting PPM with language-independent acoustic modeling. In Chapter 2, DNNs via the supervised monolingual ASR training have been used in support of producing the phonetic event streams as PPM search index. Another natural alternative is to use the universal phone set ASR from Chapter 3 to produce the phone/triphone posteriorgrams for the phonetic event selection. Going forward, we aim to enables PPM's viability in language-independent processing. Additionally, [127, 128] showed that the mismatched crowdsourcing in which nonspeakers of the language write what they hear could provide useful probabilistic transcripts, and cross-lingual ASR adaptation on such noisy transcripts has demonstrated improved phoneme error rates. This suggests a way to improve the universal phone modeling for a new language of interest.

Weakly supervised learning for AUD. In Chapter 4, both lexical and phonetic discovery approaches have been examined but the synergies between the two have yet to be explored. [129, 130] similarly assumed that repetitions of the same word shared the same or similar sequence of subword units. Thus [130] used the GMM-HMM based AUD models to decode pairs of repeated words, and constrained the two acoustic unit sequences decoded from each word pair to be similar, which demonstrated marginal NMI improvements in AUD performance. Toward this end, we would also suggest that the word pairs detected via UTD could provide weak but useful supervision information for the unsupervised acoustic model learning, and the state-of-the-art

deep generative AUD might likely benefit more from it. Vice versa, the improved acoustic unit sequences can also aid in different UTD approaches by being segmented into terms [131, 108], or by providing initial speech segmentation for the subsequent clustering process [132].

Using AUD for spoken document retrieval. [102] has showed that UTD could be useful in the ranked retrieval of spoken documents, without the need for traditional transcription or ASR. A promising next step is to look to if the n-gram acoustic units identified by AUD can be similarly effective indexing units for the same or similar retrieval tasks, and provide complementary use. Also, the search engine presented in [103] demonstrated UTD could facilitate corpus exploration by linking similar content in different recordings, and we could instead consider AUD to be applicable to the same functionality.

Bibliography

[1] R. Baeza-Yates and B. Ribeiro-Neto, "Modern Information Retrieval: the Concepts and Technology behind Search (second edition)," *Addison-Wesley*, 2011.

[2] C. Chelba, T. J. Hazen, B. Ramabhadran, and M. Saraçlar, "Speech retrieval," in *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, G. Tur and R. De Mori, Eds. John Wiley & Sons, 2011, ch. 15, pp. 417–446.

[3] L.-s. Lee, J. Glass, H.-y. Lee, and C.-a. Chan, "Spoken content retrieval–beyond cascading speech recognition with text retrieval," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1389–1420, 2015.

[4] C. D. Manning, P. Raghavan, and H. Schütze, "Introduction to Information Retrieval," *Cambridge University Press*, 2008.

[5] J. S. Garofolo, C. G. Auzanne, and E. M. Voorhees, "The TREC spoken document retrieval track: A success story," in *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC-8)*, 1999.

[6] J. G. Fiscus and G. R. Doddington, "Topic detection and tracking evaluation overview," in *Topic Detection and Tracking: Event-based Information Organization*, J. Allan, Ed. Springer Science & Business Media, 2002, ch. 2, pp. 17–31.

[7] C. L. Wayne, "Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation," in *Proc. Language Resources and Evaluation Conference (LREC)*, 2000.

[8] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, "Results of the 2006 Spoken Term Detection Evaluation," in *Proc. SIGIR Workshop on Searching Spontaneous Conversational Speech*, 2007.

[9] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proc. ACM SIGIR conference on Research and development in information retrieval*, 2007.

[10] M. Harper, "IARPA Babel Program," https://www.iarpa.gov/index.php/research-programs/babel, 2014, [Online; accessed Sep-2018].

[11] V. Manohar, D. Povey, and S. Khudanpur, "Semi-supervised maximum mutual information training of deep neural network acoustic models," in *Proc. Interspeech*, 2015.

[12] K. M. Knill, M. J. Gales, A. Ragni, and S. P. Rath, "Language independent and unsupervised acoustic models for speech recognition and keyword spotting," in *Proc. Interspeech*, 2014.

[13] J. Trmal, M. Wiesner, V. Peddinti, X. Zhang, P. Ghahremani, Y. Wang, V. Manohar, H. Xu, D. Povey, and S. Khudanpur, "The Kaldi OpenKWS system: Improving low resource keyword search," in *Proc. Interspeech*, 2017.

[14] S. Strassel and J. Tracey, "LORELEI language packs: Data, tools, and resources for technology development in low resource languages," in *Proc. LREC*, 2016.

[15] S. M. Strassel, A. Bies, and J. Tracey, "Situational awareness for low resource languages: the LORELEI situation frame annotation task," in *Proceedings of the first workshop on: Exploitation of Social Media for Emergency Relief and Preparedness (SMERP)*, 2017.

[16] M. Wiesner, C. Liu, L. Ondel, C. Harman, V. Manohar, J. Trmal, Z. Huang, N. Dehak, and S. Khudanpur, "Automatic speech recognition and topic identification for almost-zero-resource languages," in *Proc. Interspeech*, 2018.

[17] M. Mohri, F. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in *Springer Handbook of Speech Processing*. Springer, 2008, pp. 559–584.

[18] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karafiát, S. Kombrink, P. Motlíček, Y. Qian *et al.*, "Generating exact lattices in the WFST framework," in *Proc. ICASSP*, 2012.

[19] M. Hannemann, "Finite-state based recognition networks for forward-backward speech decoding," Ph.D. dissertation, Brno University of Technology, 2016.

[20] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[21] C. Chelba, J. Silva, and A. Acero, "Soft indexing of speech content for search in spoken documents," *Computer Speech & Language*, vol. 21, no. 3, pp. 458–478, 2007.

[22] A. Jansen and P. Niyogi, "Point process models for spotting keywords in continuous speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 8, pp. 1457–1470, 2009.

[23] K. Kintzley, A. Jansen, and H. Hermansky, "MAP estimation of whole-word acoustic models with dictionary priors," in *Proc. Interspeech*, 2012.

[24] ——, "Featherweight phonetic keyword search for conversational speech," *Proc. ICASSP*, 2014.

[25] ——, "Event selection from phone posteriorgrams using matched filters," in *Proc. Interspeech*, 2011.

[26] K. Kintzley, A. Jansen, K. Church, and H. Hermansky, "Inverting the point process model for fast phonetic keyword search," in *Proc. Interspeech*, 2012.

[27] A. Jansen and P. Niyogi, "Detection-based speech recognition with sparse point process models," in *Proc. ICASSP*, 2010.

[28] C. Liu, A. Jansen, G. Chen, K. Kintzley, J. Trmal, and S. Khudanpur, "Low-resource open vocabulary keyword search using point process models," in *Proc. Interspeech*, 2014.

[29] C. Liu, A. Jansen, and S. Khudanpur, "Context-dependent point process models for keyword search and detection-based ASR," in *Proc. ICASSP*, 2016.

[30] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, 1994, pp. 307–312.

[31] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 30–42, 2012.

[32] P. Fousek and H. Hermansky, "Towards ASR based on hierarchical posterior-based keyword recognition," in *Proc. ICASSP*, 2006.

[33] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[34] J. Trmal, G. Chen, D. Povey, S. Khudanpur, P. Ghahremani, X. Zhang, V. Manohar, C. Liu, A. Jansen, D. Klakow *et al.*, "A keyword search system using open source software," in *Proc. SLT*, 2014.

[35] K. Kintzley, "Phonetic event-based whole-word modeling approaches for speech recognition," Ph.D. dissertation, The Johns Hopkins University, 2014.

[36] M. J. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[37] G. S. Sivaram and H. Hermansky, "Multilayer perceptron with sparse hidden outputs for phoneme recognition," in *Proc. ICASSP*, 2011.

[38] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *the Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[39] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *Proc. ICASSP*, 2014.

[40] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Proc. ICASSP*, 2014.

[41] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[42] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for oov keywords in the keyword search task," in *Proc. ASRU*, 2013.

[43] K. Kintzley, A. Jansen, and H. Hermansky, "Text-to-speech inspired duration modeling for improved whole-word acoustic models," in *Proc. Interspeech*, 2013.

[44] D. R. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proc. Interspeech*, 2007.

[45] G. Chen, S. Khudanpur, D. Povey, J. Trmal, D. Yarowsky, and O. Yilmaz, "Quantifying the value of pronunciation lexicons for keyword search in low resource languages," in *Proc. ICASSP*, 2013.

[46] H. Xu, D. Povey, L. Mangu, and J. Zhu, "Minimum Bayes risk decoding and system combination based on a recursion for edit distance," *Computer Speech & Language*, vol. 25, no. 4, pp. 802–828, 2011.

[47] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[48] J. Glass, T. J. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in the MIT spoken lecture processing project," in *Proc. Interspeech*, 2007.

[49] Y. Park and S. C. Gates, "Towards real-time measurement of customer satisfaction using automatically generated call transcripts," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 1387–1396.

[50] C. Liu, M. Wiesner, S. Watanabe, C. Harman, J. Trmal, N. Dehak, and S. Khudanpur, "Low-resource contextual topic identification on speech," in *Proc. SLT*, 2018.

[51] N. Malandrakis, O. Glembek, and S. Narayanan, "Extracting situation frames from non-English speech: Evaluation framework and pilot results," in *Proc. Interspeech*, 2017.

[52] "APPEN," http://appen.com.

[53] T. J. Hazen, F. Richardson, and A. Margolis, "Topic identification from audio recordings using word and phone recognition lattices," in *Proc. ASRU*, 2007.

[54] M. Dredze, A. Jansen, G. Coppersmith, and K. Church, "NLP on spoken documents without ASR," in *Proc. EMNLP*, 2010.

[55] J. Wintrode and S. Khudanpur, "Limited resource term detection for effective topic identification of speech," in *Proc. ICASSP*, 2014.

[56] C. May, F. Ferraro, A. McCree, J. Wintrode, D. Garcia-Romero, and B. Van Durme, "Topic identification and discovery on text and speech," in *Proc. EMNLP*, 2015.

[57] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, p. 391, 1990.

[58] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[59] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. ICML*, 2008.

[60] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[61] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[62] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. EMNLP*, 2013.

[63] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proc. EMNLP*, 2015.

[64] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *Proc. ICLR*, 2017.

[65] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[66] P. Papadopoulos, R. Travadi, C. Vaz, N. Malandrakis, U. Hermjakob, M. P. Pourdamghani, B. Zhang, X. Pan, D. Lu, Y. Lin *et al.*, "Team ELISA system for DARPA LORELEI speech evaluation 2016," in *Proc. Interspeech*, 2017.

[67] C. Liu, J. Trmal, M. Wiesner, C. Harman, and S. Khudanpur, "Topic identification for speech without ASR," in *Proc. Interspeech*, 2017.

[68] P. Xu and R. Sarikaya, "Contextual domain classification in spoken language understanding systems using recurrent neural network," in *Proc. ICASSP*, 2014.

[69] C. Liu, P. Xu, and R. Sarikaya, "Deep contextual language understanding in spoken dialogue systems," in *Proc. Interspeech*, 2015.

[70] C. Hori, T. Hori, S. Watanabe, and J. R. Hershey, "Context sensitive spoken language understanding using role dependent LSTM layers," in *Proc. NIPS Workshop on Machine Learning for Spoken Language Understanding and Interaction*, 2015.

[71] N. T. Vu, F. Kraus, and T. Schultz, "Cross-language bootstrapping based on completely unsupervised training using multilingual A-stabil," in *Proc. ICASSP*, 2011.

[72] T. Schultz, "GlobalPhone: a multilingual speech and text database developed at Karlsruhe University," in *Seventh International Conference on Spoken Language Processing*, 2002.

[73] J. C. Wells, "Computer-coding the IPA: a proposed extension of SAMPA," *Revised draft*, vol. 4, no. 28, p. 1995, 1995.

[74] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech*, 2015.

[75] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Proc. Interspeech*, 2016.

[76] V. Manohar, D. Povey, and S. Khudanpur, "JHU Kaldi System for Arabic MGB-3 ASR challenge using diarization, audio-transcript alignment and transfer learning," in *Proc. ASRU*, 2017.

[77] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[78] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for SVM," in *Proc. ICML*, 2007.

[79] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[80] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Proc. ICLR*, 2015.

[81] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," *arXiv preprint arXiv:1412.1602*, 2014.

[82] S. Khurana and A. Ali, "QCRI advanced transcription system (QATS) for the Arabic multi-dialect broadcast media recognition: MGB-2 challenge," in *Proc. SLT*, 2016.

[83] "1997 Spanish Broadcast News Speech (HUB4-NE)," https://catalog.ldc.upenn.edu/LDC98S74, [Online; accessed Sep-2018].

[84] "GALE Phase 2 Chinese Broadcast News Speech," https://catalog.ldc.upenn.edu/LDC2013S08, [Online; accessed Sep-2018].

[85] V. Peddinti, G. Chen, D. Povey, and S. Khudanpur, "Reverberation robust acoustic modeling using i-vectors with time delay neural networks," in *Proc. Interspeech*, 2015.

[86] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[87] P. Liang, B. Taskar, and D. Klein, "Alignment by agreement," in *Proc. NAACL HLT*, 2006.

[88] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media Inc., 2009.

[89] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *The International Conference on Learning Representations (ICLR)*, 2015.

[90] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," 2017.

[91] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[92] A. Das, J. Li, R. Zhao, and Y. Gong, "Advancing connectionist temporal classification with attention modeling," *Proc. ICASSP*, 2018.

[93] C. Liu, J. Yang, M. Sun, S. Kesiraju, A. Rott, L. Ondel, P. Ghahremani, N. Dehak, L. Burget, and S. Khudanpur, "An empirical evaluation of zero resource acoustic unit discovery," in *Proc. ICASSP*, 2017.

[94] Wikipedia: The Free Encyclopedia, "Orthography," https://en.wikipedia.org/wiki/Orthography, [Online; accessed Sep-2018].

[95] N. Coupland, *The handbook of language and globalization*. John Wiley & Sons, 2011, vol. 64.

[96] M.-h. Siu, H. Gish, A. Chan, W. Belfield, and S. Lowe, "Unsupervised training of an HMM-based self-organizing unit recognizer with applications to topic classification and keyword discovery," *Computer Speech & Language*, vol. 28, no. 1, pp. 210–223, 2014.

[97] S. Kesiraju, R. Pappagari, L. Ondel, L. Burget, N. Dehak, S. Khudanpur, J. Černockỳ, and S. Gangashetty, "Topic identification of spoken documents using unsupervised acoustic unit discovery," in *Proc. ICASSP*, 2017.

[98] H. Kamper, A. Jansen, and S. Goldwater, "Unsupervised word segmentation and lexicon discovery using acoustic word embeddings," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 4, pp. 669–679, 2016.

[99] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, 2008.

[100] A. Jansen and B. Van Durme, "Efficient spoken term discovery using randomized algorithms," in *Proc. ASRU*, 2011, https://github.com/arenjansen/ZRTools.

[101] V. Lyzinski, G. Sell, and A. Jansen, "An evaluation of graph clustering methods for unsupervised term discovery," in *Proc. Interspeech*, 2015.

[102] J. White, D. Oard, A. Jansen, J. Paik, and R. Sankepally, "Using zero-resource spoken term discovery for ranked retrieval," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 588–597.

[103] D. W. Oard, R. Sankepally, J. White, and C. Harman, "Vapor Engine: Demonstrating an early prototype of a language-independent search engine for speech," in *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*. ACM, 2016, pp. 301–304.

[104] A. Jansen, E. Dupoux, S. Goldwater, M. Johnson, S. Khudanpur, K. Church, N. Feldman, H. Hermansky, F. Metze, R. Rose *et al.*, "A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition," in *Proc. ICASSP*, 2013.

[105] C.-y. Lee and J. Glass, "A nonparametric bayesian approach to acoustic model discovery," in *Proc. ACL*, 2012.

[106] L. Ondel, L. Burget, and J. Černocký, "Variational inference for acoustic unit discovery," in *Proc. SLTU*, 2016.

[107] J. Ebbers, L. D. Jahn Heymann, T. Glarner, R. Haeb-Umbach, and B. Raj, "Hidden markov model variational autoencoder for acoustic unit discovery," in *Proc. Interspeech*, 2017.

[108] L. Ondel, P. Godard, L. Besacier, E. Larsen, M. Hasegawa-Johnson, O. Scharenborg, E. Dupoux, L. Burget, F. Yvon, and S. Khudanpur,

"Bayesian models for unit discovery on a very low resource language," in *Proc. ICASSP*, 2018.

[109] T. Glarner, P. Hanebrink, J. Ebbers, and R. Haeb-Umbach, "Full Bayesian hidden Markov model variational autoencoder for acoustic unit discovery," in *Proc. Interspeech*, 2018.

[110] T. J. Hazen, "MCE Training Techniques for Topic Identification of Spoken Audio Documents," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2451–2460, Nov 2011.

[111] P. Xu and R. Sarikaya, "Convolutional neural network based triangular CRF for joint intent detection and slot filling," in *Proc. ASRU*, 2013.

[112] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[113] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[114] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[115] S. Semeniuta, A. Severyn, and E. Barth, "A hybrid convolutional variational autoencoder for text generation," in *Proc. EMNLP*, 2017.

[116] C.-H. Shen, J. Y. Sung, and H.-Y. Lee, "Language transfer of audio word2vec: Learning audio segment representations without target language data," in *Proc. ICASSP*, 2018.

[117] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, 1993.

[118] J. Godfrey, E. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. ICASSP*, 1992.

[119] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[120] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820*, 2015.

[121] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[122] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016.

[123] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[124] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv*, 2012.

[125] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 2010, pp. 45–50.

[126] "VoxForge," http://www.voxforge.org, [Online; accessed Sep-2018].

[127] C. Liu, P. Jyothi, H. Tang, V. Manohar, R. Sloan, T. Kekona, M. Hasegawa-Johnson, and S. Khudanpur, "Adapting ASR for under-resourced languages using mismatched transcriptions," in *Proc. ICASSP*, 2016.

[128] M. A. Hasegawa-Johnson, P. Jyothi, D. McCloy, M. Mirbagheri, G. M. d. Liberto, A. Das, B. Ekin, C. Liu, V. Manohar, H. Tang *et al.*, "ASR for under-resourced languages from probabilistic transcription," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 25, no. 1, pp. 50–63, 2017.

[129] A. Jansen, S. Thomas, and H. Hermansky, "Weak top-down constraints for unsupervised acoustic model training," in *Proc. ICASSP*, 2013.

[130] S. Shum, "Overcoming resource limitations in the processing of unlimited speech: applications to speaker and language recognition," Ph.D. dissertation, Massachusetts Institute of Technology, 2016.

[131] C.-y. Lee, T. J. O'Donnell, and J. Glass, "Unsupervised lexicon discovery from acoustic input," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 389–403, 2015.

[132] S. Bhati, H. Kamper, and K. S. R. Murty, "Phoneme based embedded segmental k-means for unsupervised term discovery," in *Proc. ICASSP*, 2018.

Vita

Chunxi Liu received the B.Sc. degree in Electronic Information Engineering from the Harbin Institute of Technology, Harbin, China, and the B.Eng. degree in Communications Systems Engineering from the University of Birmingham, Birmingham, U.K., both in 2012. He enrolled in the Ph.D. program in the Department of Electrical and Computer Engineering at the Johns Hopkins University in August 2012, and has been a member of the Center for Language and Speech Processing since then. His research interests include acoustic modeling for automatic speech recognition and spoken language understanding. He worked as an intern in the Speech and Language Sciences Group at Microsoft Research during the summer of 2014 . He received a Speech and Language Processing Student Paper Award at the IEEE International Conference on Acoustics, Speech and Signal Processing, 2016. His paper was shortlisted for the Best Student Paper Award at the INTERSPEECH conference, 2017.